Robust detection of astronomical sources using convolutional neural networks

Maxime Paillassa

université de BORDEAUX

THÈSE PRESENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR DE L'UNIVERSITÉ DE BORDEAUX

École doctorale 209 : Sciences Physiques et de l'Ingénieur

Spécialité "Astrophysique, plasmas, nucléaire"

Par Maxime PAILLASSA

Détection robuste de sources astronomiques par réseaux de neurones à convolutions

Sous la direction de Hervé BOUY et Emmanuel BERTIN

Le 16 Octobre 2020

Membres du jury:

M. BORDÉ Pascal,	Professeur, Université de Bordeaux	Président du jury
M. HUERTAS-COMPANY Marc,	Maître de conférence, Université de Paris	Rapporteur
M. AUBOURG Éric,	Directeur de recherches, CEA	Rapporteur
Mme. E.O. ISHIDA Emille,	Ingénieur de recherche, CNRS	Examinateur
M. GIOVANNELLI Jean-François,	Professeur, Université de Bordeaux	Examinateur
Mme. MÖLLER Anais,	Chargée de recherche, CNRS	Examinateur
M. DABIN Christophe,	Chef de Projet, CNES	Invité
M. CUCCHETTI Edoardo,	Ingénieur qualité d'image, CNES	Invité

Abstract

Robust detection of astronomical sources using convolution neural networks

Extracting reliable source catalogs from images is crucial for a broad range of astronomical research topics. However, the efficiency of current source detection methods becomes severely limited in crowded fields, or when images are contaminated by optical, electronic and environmental defects. Performance in terms of reliability and completeness is now often insufficient with regard to the scientific requirements of large imaging surveys. In this thesis, we develop new methods to produce more robust and reliable source catalogs. We leverage recent advances in deep supervised learning to design generic and reliable models based on convolutional neural networks (CNNs). We present MaxiMask and MaxiTrack, two convolutional neural networks that we trained to automatically identify 13 different types of image defects in astronomical exposures. We also introduce a prototype of a multi-scale CNN-based source detector robust to image defects, which we show to significantly outperform existing algorithms. We discuss the current limitations and potential improvements of our approach in the scope of forthcoming large scale surveys such as Euclid.

Keywords: Image processing - Deep learning - Convolutional neural networks - Source detection - Wide field surveys

---

Research unit
Laboratoire d'Astrophysique de Bordeaux
Université de Bordeaux & CNRS (UMR 5804)
Univ. Bordeaux – Bât. B18N, allée Geoffroy Saint-Hilaire, CS 50023, 33615 Pessac Cedex, France

Résumé

Détection robuste de sources astronomiques par réseaux de neurones à convolutions

L'extraction de catalogues de sources fiables à partir des images est cruciale pour un large éventail de recherches en astronomie. Cependant, l'efficacité des méthodes de détection de source actuelles est sérieusement limitée dans les champs encombrés, ou lorsque les images sont contaminées par des défauts optiques, électroniques et environnementaux. Les performances en termes de fiabilité et de complétude sont aujourd'hui souvent insuffisantes au regard des exigences scientifiques des grands relevés d'imagerie. Dans cette thèse, nous développons de nouvelles méthodes pour produire des catalogues sources plus robustes et fiables. Nous tirons parti des progrès récents en apprentissage supervisé profond pour concevoir des modèles génériques et fiables basés sur des réseaux de neurones à convolutions (CNNs). Nous présentons MaxiMask et MaxiTrack, deux réseaux de neurones à convolutions que nous avons entrainés pour identifier automatiquement 13 types différents de défauts d'image dans des expositions astronomiques. Nous présentons également un prototype de détecteur de sources multi-échelle et robuste vis-à-vis des défauts d'image, dont nous montrons qu'il surpasse largement les algorithmes existants en terme de performances. Nous discutons des limites actuelles et des améliorations potentielles de notre approche dans le cadre des prochains grands relevés tels que Euclid.

Mots clés: Traitement d'images - Apprentissage profond - Réseaux de neurones à convolutions - Détection de sources - Relevés grand champ

---

Unité de recherche
Laboratoire d'Astrophysique de Bordeaux
Université de Bordeaux & CNRS (UMR 5804)
Univ. Bordeaux – Bât. B18N, allée Geoffroy Saint-Hilaire, CS 50023, 33615 Pessac Cedex, France

Remerciements

Même si j'ai toujours eu une certaine fascination pour l'Univers et ses mystères, mon arrivée dans un laboratoire d'Astrophysique a été plutôt tardive. Au moment de mes choix de parcours, ma curiosité des ordinateurs l'avait emporté, m'amenant alors dans une école d'ingénieur en informatique. C'est pourquoi je tiens à remercier en tout premier lieu Emmanuel Bertin, qui m'a donné l'opportunité de mettre un pied en Astrophysique en m'acceptant en stage de fin d'études à l'Institut d'Astrophysique de Paris. Sans ce point de départ, je n'aurais sûrement jamais eu la chance de faire cette thèse. La réalisation de ce projet est aussi le fruit d'efforts de mes directeurs de thèse, Hervé Bouy et Emmanuel Bertin, ainsi que de membres du CNES, cofinanceur de la thèse. Merci à vous tous d'avoir soutenu ce sujet et permis à cette thèse d'avoir lieu.

Ce fut une réelle chance pour un non astronome de passer ces trois riches années dans un laboratoire d'Astrophysique. Une grande partie de cette richesse provient évidemment de mes directeurs de thèse, Hervé Bouy et Emmanuel Bertin. J'espère pouvoir garder et retenir de vous vos ambitions passionnées, votre rigueur et votre bonne humeur qui furent une grande source de motivation. Merci pour toutes les connaissances que vous avez su apporter à un non astronome et merci de m'avoir toujours incité et permis d'aller présenter mes travaux à des conférences, workshops, etc.

Un grand merci à Hervé pour m'avoir permis de faire des observations à l'Observatoire de Haute-Provence et à La Palma (sans avoir eu à faire de proposal!). J'ai ainsi pu découvrir ce monde fascinant et me sentir un peu plus astronome! J'en profite pour remercier ici tous mes acolytes d'observation et tous les personnels techniques des observatoires qui ont oeuvré au bon déroulement de ces observations.

Je tiens aussi à remercier les personnels administratifs du laboratoire pour la préparation et la gestion des mes (nombreuses) missions.

Je remercie l'ensemble des membres du jury pour avoir accepté d'examiner mes travaux et pour m'avoir donné de précieuses idées quant à l'évolution possible de ces travaux.

Merci à toutes les personnes, membres du laboratoire ou non, que j'ai pu cotôyer et qui ont embelli ces trois années. Je ne citerai pas de nom pour éviter les oublis, les concernés se reconnaîtront forcément! Je pense notamment aux pauses cafés qui s'éternisent, aux séances de jeux de société, au Swing Marine, aux escape games, aux séances de statistiques..., au soirées quizz du Sherlock et du Titi Twister, au Hellfest, au Teppanyaki, au Blarney Stone, au week-end ski, aux concerts, aux squats de bureaux, aux sessions de bloc, aux longues discussions, et je dois certainement en oublier!

Je pense aussi bien évidemment au théâtre, au confinement, à la cabane, aux balades (avec puis sans lunettes...), à l'accrobranche, à la traverséé de la France d'Ouest en Est, et j'en passe! Malgré les circonstances, elle ont fait de cette troisième année une des plus belles.

Finalement, je tiens à remercier mon père, ma mère et ma soeur, qui me soutiennent toujours, peu importe les directions que je prends dans ma vie.

Contents

Chapter 1

Introduction

Much of the science carried out in Astrophysics depends on source catalogs. The vast majority of astronomical sources cataloged so far have been detected in wide-field images taken at visible and Near-InfraRed (NIR) wavelengths. Source detection is a crucial stage in the exploitation of imaging data, especially in large sky photometric surveys. However, the current detection performance in terms of reliability and completeness is now insufficient with regard to the scientific requirements of current and forthcoming experiments, e.g., HSC (Aihara et al., 2018), Euclid (Racca et al., 2016), or LSST (Ivezić et al., 2019). A performance leap is necessary, which must also satisfy the processing time constraints imposed by the large amount of data to be processed.

In this context, we aim to design the most universal possible source detector for optical and NIR wide-field instruments. By universal, we mean that it must be able to work with various telescopes, cameras and observing conditions without requiring extensive tuning. We also aim to make a robust detector regarding whatever defects or imperfections may affect images.

This project was initiated in the context of two particular surveys: Cosmic-DANCe (Bouy et al., 2017), standing for Dynamical Analysis of Nearby Clusters, hereafter written COSMIC-DANCE, and Euclid (Laureijs et al., 2012).

The primary goal of the COSMIC-DANCE survey is to recover the initial stellar mass function, i.e., the function describing the formation rate of stars as a function of mass, by studying young and nearby open clusters. COSMIC-DANCE focuses on the lowest mass stars, below the Gaia (Gaia Collaboration et al., 2016) magnitude limit. This population is poorly known because of the high contamination and incompleteness rates in this observation regime. COSMIC-DANCE gathers wide-field imaging data of nearby open clusters and star-forming regions from a large range of ground-based observations and data archives. These data are used to compile star catalogs with proper motion measurements, i.e., the apparent motion of stars in the sky, and cluster membership probabilities, i.e., the probability for a star to belong to the cluster. It is thus critical for COSMIC-DANCE to have a universal source detection tool, capable of handling the wide heterogeneity of the data to be processed. Robust and reliable tools are also required to manage the unknown and variable image quality of the data retrieved from the archives.

The Euclid mission relies on a space-based telescope developed and operated by ESA, with both optical and NIR wide-field cameras onboard. Euclid primarily aims at understanding the nature of dark matter and dark energy by measuring precisely the accelerated expansion of the Universe. Weak galaxy lensing (the tiny distortion of galaxy shapes due to the deviation of light rays by massive structures along the line of sight) and galaxy clustering are two major probes of dark matter and dark energy used by Euclid. Therefore, the robust detection of galaxies and the precise estimation of their positions and shapes are among the strong requirements of the Euclid mission (Amiaux et al., 2010).

In addition to these particular surveys, many upcoming surveys plan to gather tremendous

amounts of data, making the design of reliable, robust, automatic and fast source detection tools necessary.

In practice one must distinguish point-like sources, i.e., stars and quasars, from extended sources, which are mainly galaxies but can also include compact nebulae or stellar clusters whose stars are unresolved. Currently, methods exist that are known to be optimal for detecting isolated point sources, such as algorithms based on the matched filter (Woodward, 1953, 2014; Bertin and Arnouts, 1996). Yet, the efficiency of these methods becomes severely limited in crowded fields (i.e., when the source density is so high that source images overlap, a phenomenon known as blending), or when images are contaminated by optical, electronic and environmental defects. Some limitations of matched filter-based detection in such regimes are shown in Fig. 1.1.

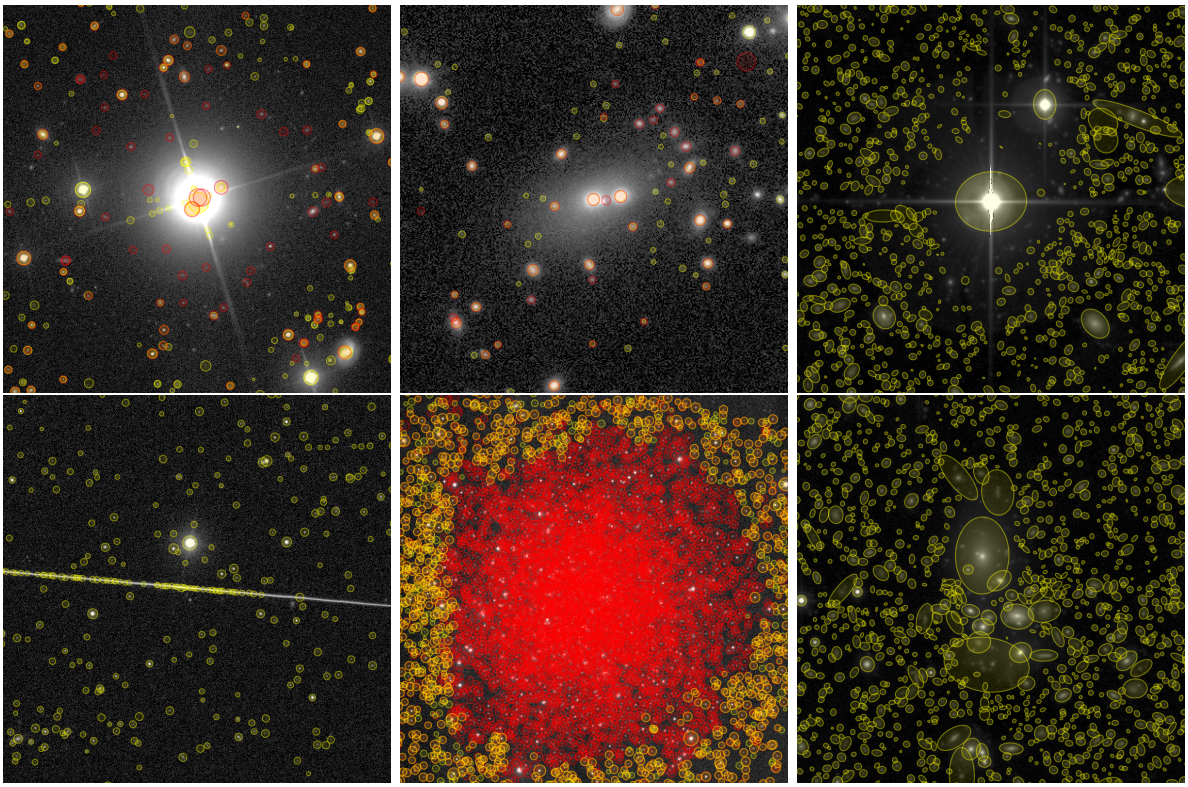


Figure 1.1: Illustrations of the main limitations of current source detection algorithms. Yellow circles are SDSS detections (12th data release, Alam et al., 2015) while red circles are Pan-STARRS detections (1st data release, Flewelling, 2017, 2018), excepted for the right images which present SExtractor detections in CFHTLS (Cuillandre and Bertin, 2006). Left images show contaminant issues. Top left: false detections on the star diffraction spikes and saturated core. Note also how sources around the bright star are missed. Bottom left: false detections on a trail crossing the image. Middle images show deblending issues. Top middle: the neighbor of the central source is not well detected in Pan-STARRS and even missed in the SDSS. Bottom middle: Pan-STARRS over deblending in the NGC 5466 globular cluster while the SDSS simply ignores this area. Right images show both contaminant and deblending issues. Top right: sources around the bright star are ignored. Bottom right: sources around the most extended ones are not detected. Images are seen through Visiomatic 2 (Bertin et al., 2019a).

Stellar crowding is particularly problematic in low-galactic latitude fields, where confusion noise defines the detection limit and largely dominates the photometric and astrometric error budgets. The situation is more severe in the NIR domain, where extinction due to interstellar dust is significantly reduced. As of 2020 the best performing methods in crowded images are still

largely empirical and consist in iteratively subtracting point-sources, from the brightest to the faintest, using a Point Spread Function (PSF) model (Stetson, 1987; Schechter et al., 1993; Zhang and Kainulainen, 2019). The detection of the faintest stars is also complicated by the presence of contaminants. The most troublesome contaminants are: optical ghosts, especially in wide-field cameras; cosmic rays and hot pixels in under-sampled images; and nebulae. With current algorithms, the smaller and sharpest contaminated areas can be recovered through interpolation (Popowicz et al., 2013), provided they have been formerly identified. On the other hand, the most extended contaminants like nebulae are currently handled with complex background estimation techniques (Popowicz and Smolka, 2015) or Bayesian models Knollmüller et al. (2018). In this context, having reliable and versatile tools to detect and manage contaminants is essential.

Unlike stars that are point-like sources, galaxies appear extended. For extended objects detection completeness does not only depend on magnitude, i.e., total flux, but also on surface brightness, i.e., the measure of brightness per unit of detector area (or solid angle). The detection selection function of galaxies is thus two dimensional (Driver et al., 2005), as shown in Fig. 1.2. Even when isolated, low surface brightness galaxies can easily be missed by simple thresholding algorithms operating at a single detection scale, as Fig. 1.2 shows.

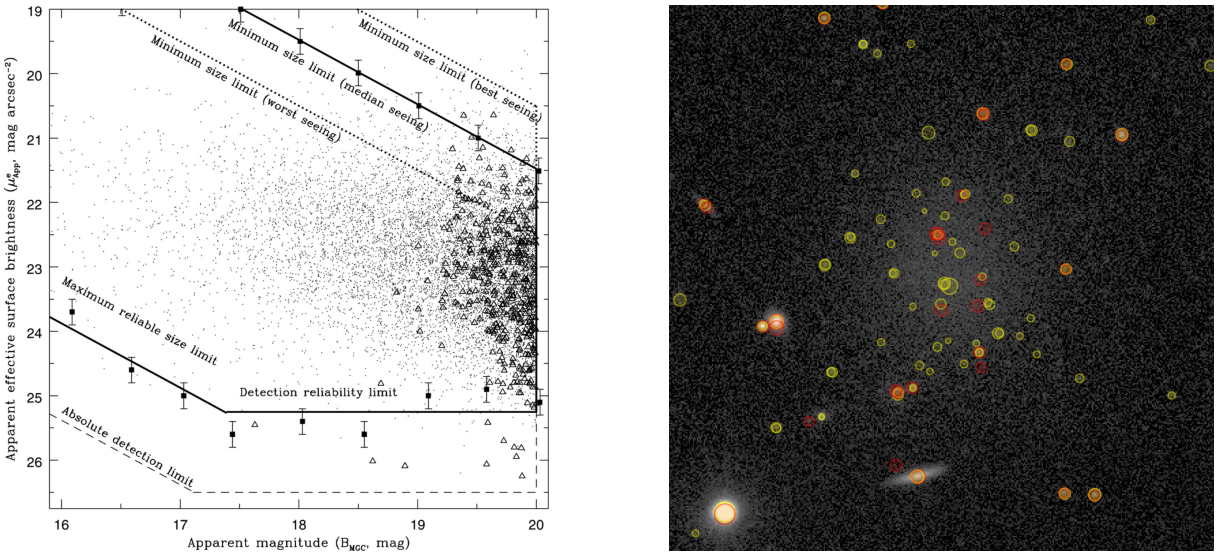


Figure 1.2: Left: simplified view of the galaxy detection selection function in the Millennium galaxy catalog (Driver et al., 2005). Because galaxies are extended sources, there is a detection limit in magnitude and a detection limit in surface brightness. The slanted bottom left line defines a maximal size detection limit: if a galaxy is too extended, it cannot be detected. The slanted top right lines define minimal size detection limits: if a galaxy is too small, it can be confused with a point-like source, especially with poor seeings. Right: an example of missed low surface brightness galaxy in the SDSS and Pan-STARRS catalogs. There are even some noise peak false detections within the galaxy. The right image is seen through Visiomatic 2 (Bertin et al., 2019a).

Yet, those objects are of great importance in astrophysics. Indeed, according to observational cosmology, the most prominent galaxy formation scenarios derived from the cold dark matter model predict that such objects are abundant, in the form of satellite galaxies or in "pearled" galaxy filaments, both dominated by dark matter (Kauffmann et al., 1993; Moore et al., 1999). Though, apart from the low surface brightness, the detection of such objects is also complicated by the presence of intra-cluster light (Contini et al., 2014), galaxy collision residuals such as shells, tails or stream structures (Hendel and Johnston, 2015), and diffuse galactic cirrus from

cold dust clouds (Miville-Deschênes et al., 2016). Illustrations of such features are shown in Fig. 1.3.

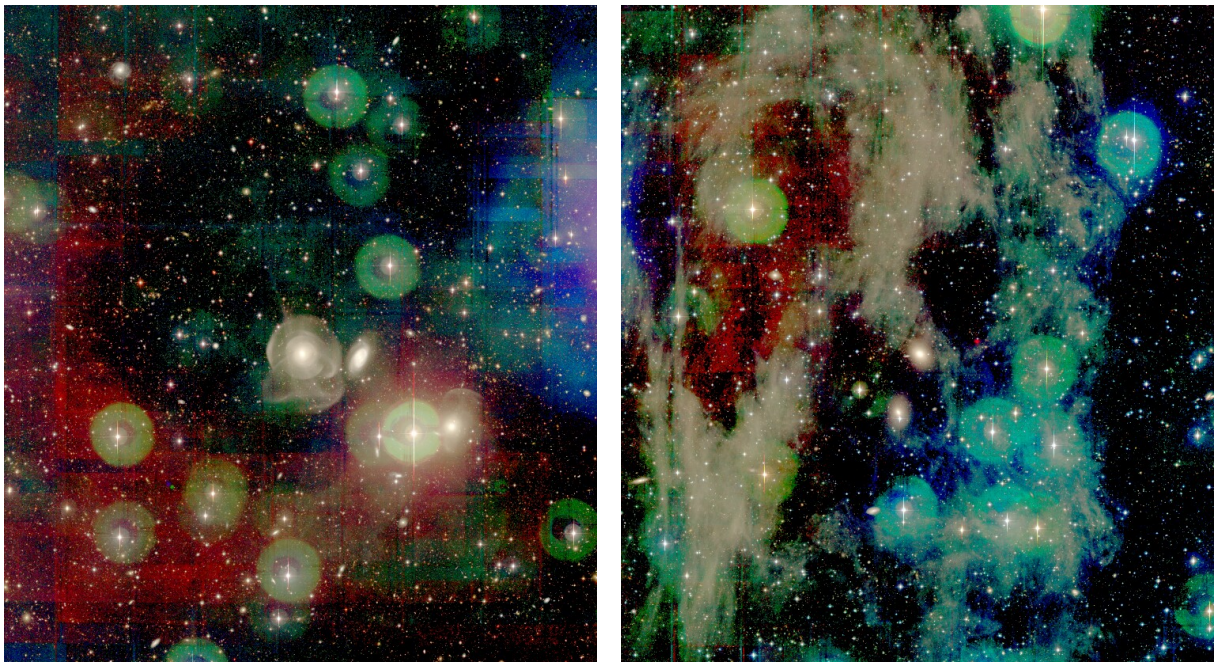


Figure 1.3: Example of galaxy images from the MATLAS survey (Duc et al., 2015) illustrating low surface brightness artifacts. Left: NGC 0474. The galaxy shows several shells and radial streams. Right: NGC 2592. Cirrus is spread over the field. Other images are available at http://irfu.cea.fr/Projets/matlas/public/Atlas3D/atlas3d-pXXIX.html. Images are seen through Visiomatic 2 (Bertin et al., 2019a).

A lot of efforts are put into the inventory and measurements of these objects through current or upcoming experiments, including Dragonfly (Abraham and van Dokkum, 2014), Messier (Valls-Gabaud and MESSIER Collaboration, 2017), Huntsman (Spitler et al., 2019), MATLAS (Duc, 2020) and CASTLE (Lombardo et al., 2020). One may consider increasing the detectability of these objects by leveraging multi-scale approaches (Starck et al., 2000). However, besides the aforementioned physical objects, many low surface brightness contaminants interfere with the detection of these galaxies in practice, including star halos and ghosts, fringing and flat field residuals. As no automatic algorithms capable of working in this regime are available in detection pipelines, visual inspection remains necessary (Bílek et al., 2020). It is thus essential to develop multi-scale detection algorithms that are "intelligent" enough to handle such complex situations.

Furthermore, crowding also affects galaxy detection and measurements. Galaxies are not distributed independently across the sky. Through the action of gravity on primordial Universe density fluctuations, they distribute in clusters, sheets and filaments. Galaxy images are thus likely to overlap, or even blend, possibly with foreground stars. This blending strongly affects the statistics derived from galaxy catalogs, in particular in observational cosmology, e.g., when measuring the galaxy correlation function, cluster richness (Gruen et al., 2019) or weak lensing magnifications (Gaztanaga et al., 2020). This is how approximately 20% of the sources identified as galaxies in the deepest ground-based survey catalogs end up being removed from weak lensing measurement datasets because of blending (Chang et al., 2013), even though this mainly concerns source measurements and not detections. At the detection level, the statistic biases caused by blending can be estimated using image simulations (Chang et al., 2015; Suchyta et al., 2016). There is even good hope to get free from these biases with Approximate Bayesian Computation

methods (Carassou et al., 2017; Kacprzak et al., 2020; Tortorelli et al., 2020).

All these current challenges and constraints will be even more important to face with upcoming large scale surveys like Euclid or the LSST. Analyzing the unprecedented amount of data obtained in these surveys requires new, fast, reliable and unattended detection tools for both sources and contaminants.

In this PhD work, we aim to exploit the new data-driven approaches that have emerged recently. We aspire to take advantage of supervised learning techniques and convolutional neural networks (LeCun et al., 1995), which have proven effective in computer vision tasks such as image classification (assigning labels to images, Krizhevsky et al., 2012; Simonyan and Zisserman, 2014), image segmentation (assigning labels to pixels, Ronneberger et al., 2015; Badrinarayanan et al., 2017), and instance-aware object detection (Redmon et al., 2016; Ren et al., 2015; He et al., 2017), where each object is individually detected and eventually segmented. This is a complete paradigm shift from more traditional algorithmic approaches, changing the way problems are addressed.

This manuscript is divided in chapters organized as follows: in Chapter 2, I introduce our model describing optical and NIR wide-field images. This will help us identify the most relevant features of astronomical images and pose the problem of source detection. After reviewing possible solutions to this problem in Chapter 3, I justify our choice of a machine learning based approach. In Chapter 4, I introduce the necessary concepts related to the supervised machine learning techniques that we apply to images: convolutional neural networks. This brings us to Chapter 5, where I tackle the identification of contaminants with MaxiMask and MaxiTrack. In Chapter 6 I focus on the problem of source detection, and present our new detector prototype based on convolutional neural networks. Finally, I provide a summary of our results and discuss future work directions in Chapter 7.

Chapter 2

Astronomical images and the source detection problem

Astronomical images are the result of several processes. Fig. 2.1 illustrates a simplified view of an astronomical observation made from the ground:

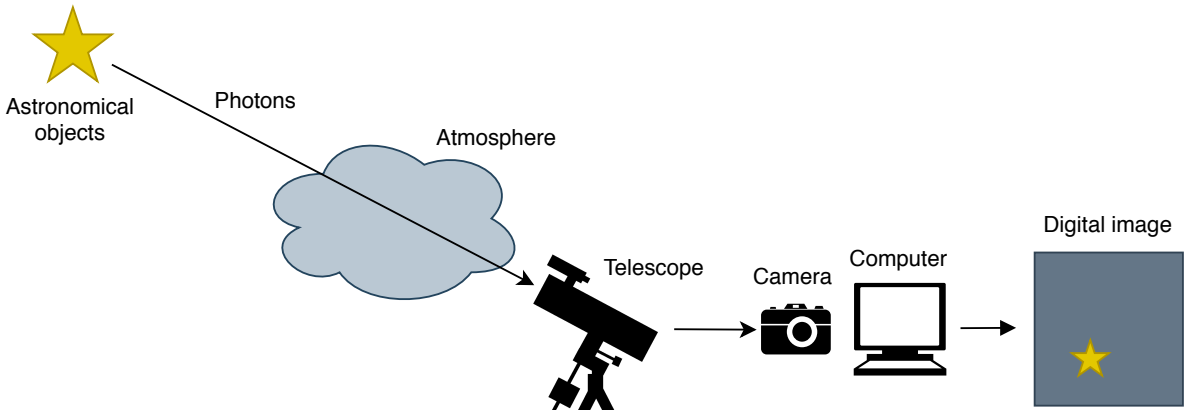


Figure 2.1: Schematic and simplified representation of a ground-based astronomical observation.

Photons emitted by astronomical objects are collected by a telescope pointing in their direction. They travel through space and the atmosphere until the optics of the telescope focus them on a camera constituted of pixels. In each pixel, they are counted and each count is converted into a number to build a digital image, i.e., an array of numbers.

Our aim is to design a universal source detector, in the sense that it should be able to adapt to various telescopes, cameras and ambient conditions. With this in mind, I first present our simplified yet generic model of optical and NIR wide-field images.

2.1 Image model

Having a good model of astronomical images is essential for several purposes, including understanding how images form, what is at stake in the source detection problem, and having the necessary information to realistically simulate astronomical image features if necessary.

As depicted in Fig. 2.1, the light emitted by the observed astronomical objects first travels through space and the atmosphere until it reaches the telescope. The image formation process in the telescope and instrument is linear and translation equivariant (at least locally). In that respect, the contributions of the various sources add up in the image and a translation in the

object plane results in a translation in the image plane. The image of the light emitted by astronomical objects can therefore be described as a linear combination of the impulse response of the system at each point: it is a convolution. In optical astronomy, the impulse response is called the Point Spread Function (PSF hereafter). Photons hitting the detector are counted and converted into numbers in each pixel. This introduces shot noise, which I describe in Section 2.3. Regarding the camera, we assume that all image pixels are independent and arranged in a homogeneous grid: they do not influence each other and they all have the same response and sensitivity. Under this assumption the resulting digital image is a regularly sampled version of the convolution of the light signal with a PSF (which now includes the intra-pixel response) up to a multiplicative conversion factor and additional readout noise. In real detectors, there may be crosstalk between pixels due to inter-pixel capacitance, so that pixels are no longer independent. Also, distinct pixels may have different sensitivities, breaking the homogeneity hypothesis. This can be partially mitigated by image calibrations seen in Section 2.6.

Within this framework, we can describe astronomical images as the regular sampling of a noisy realization of the result of the convolution of the object signals with the PSF of the instrument, as well as an additional readout noise:

$$\boldsymbol{y} = N(\text{Ш}_S(\boldsymbol{h} * \boldsymbol{x})) + \boldsymbol{n}, \tag{2.1}$$

where:

- $\boldsymbol{y}$ is the observed signal.
- $*$ denotes the convolution operator.
- $\boldsymbol{h}$ is the total PSF of the instrument, depending on the telescope optics, the atmosphere and the detector.
- $\boldsymbol{x}$ is the true signal.
- $\text{Ш}_S()$ denotes the Shah function, also known as Dirac comb, impulse train or sampling function. It samples the continuous signal into a discrete signal, where S is the sampling period and corresponds to the camera pixel size.
- $N()$ denotes a random process intrinsic to the counting of photons.
- $\boldsymbol{n}$ is the additional readout noise due to the reading of the photon counts by the camera.

I describe the convolution, the intrinsic noise, the additive readout noise and the PSF of the instrument in more details in the next sections.

2.2 Convolution operation

In the continuous domain, the convolution operation of f and g is defined as:

$$c(t) = (f * h)(t) = \int_{-\infty}^{\infty} f(t-\tau) \cdot h(\tau) d\tau = \int_{-\infty}^{\infty} f(\tau) \cdot h(t-\tau) d\tau, \tag{2.2}$$

where c is the convolution of f and h. The convolution operation is a translation invariant linear operator[1]. Reciprocally, any translation invariant linear operator is a convolution. It is

[1] Rigorously, the convolution operation is a translation equivariant linear operator.

the mathematical representation of a linear filter. Another useful property of the convolution is that it is a multiplication in the Fourier space:

$$C(k) = F(k)H(k), \tag{2.3}$$

where C, F and H are the Fourier transforms of c, f and h, respectively. Reasoning in Fourier space can provide interesting insights but more practically, this is a way to compute faster convolution operations.

In pixelated images, the image is discrete and the convolution is written:

$$I[p] = (S * H)(p) = \sum_{q \in \mathcal{P}} S[p-q] \cdot H[q] = \sum_{q=-\infty}^{\infty} S[q] \cdot H[p-q], \tag{2.4}$$

where I is the final convolved image, S the source image, H the convolution kernel which is the PSF of the instrument and $\mathcal{P}$ is the set of pixels.

Or rather:

$$I[x,y] = (S * H)[x,y] = \sum_{h<H} \sum_{w<W} S[x-w, y-h] \cdot H[w,h], \tag{2.5}$$

where W and H are the image width and height, respectively. Each new value of pixel at $[x,y]$ is recomputed as the weighted sum of the neighboring pixels by the PSF values.

A comprehensive description of the characteristics of the PSFs encountered in astronomy is given later in Section 2.4.

2.3 Noise model

Our astronomical image model features two sources of noise.

Firstly, as mentioned earlier in Section 2.1, counting photons induces shot noise originating from the discrete nature of light. It can be modeled by a Poisson distribution with parameter λ, where λ is the expected number of collected photons during a given time interval (the exposure time in our case). The Poisson distribution probability density $p_P(k)$ is defined as:

$$p_P(k) = \frac{\lambda^k}{k!} \exp(-\lambda). \tag{2.6}$$

It gives the probability that k photon(s) are collected during the exposure time. The standard deviation being $\sqrt{\lambda}$, the signal-to-noise ratio is $\sqrt{\lambda}$. When λ is large (high photon counts, long exposure times), Poisson noise can be approximated by Gaussian noise.

The second source of noise affecting astronomical images is the additive noise due to the camera readout electronics, which has a Gaussian distribution. The Gaussian distribution probability density $p_G(x)$ is defined as:

$$p_G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{2.7}$$

where μ and σ are the expected mean and standard deviation of the distribution, respectively.

An important distinction to make between these two sources of noise is that Poisson noise is intrinsic to the signal while Gaussian readout noise is not: it is simply an additive term in the image model. This makes photon noise more difficult to manage compared to readout noise. However, in situations where sources are sufficiently faint compared to the sky background, considering photon noise as additive and stationary is a good approximation. This will generally be the case for this work, as our main observational data set consists of wide band imaging in the optical and NIR domains with long exposure times. This is an important assumption as it makes the matched filter the optimal linear filter for detecting faint isolated sources as we shall see later.

2.4 Point spread function

The PSF incorporates two main components: one comes from the instrument while the other comes from the atmosphere (only in the case of ground based observations). For more details about both components, see for instance Wilson (2000, 2001) and Roddier (1981), respectively.

2.4.1 Optical instruments

Telescopes are optical instruments made of lenses and/or mirrors. A telescope has several functions:

- Light grasp: gather a maximum of light.
- Angular resolution: resolve very close or small objects.
- Magnification: zooming details that cannot be seen with the naked eye.

There are two main types of telescopes: refracting and reflecting telescopes. Most professional wide-field instruments are mounted on reflective telescopes that use large mirrors to focus light, and a combination of smaller lenses to correct for field aberrations.

The two main optical characteristics of a telescope are the diameter D of the main aperture, which sets the light grasp power and the resolution of the telescope, and the effective focal length F of the optical combination which defines the magnification (the pixel scale in arcseconds). Another useful parameter is the focal ratio (or f-ratio) F/D. The higher the f-ratio, the lower the illuminance of an extended source, i.e., the amount of light or photons received per unit area on the focal plane.

2.4.2 Diffraction and aberrations

A major part of the PSF is defined by the telescope optics. Assuming a point-source located at an infinite distance of the telescope, the optical PSF is the Fraunhofer diffraction pattern produced by the entrance pupil of the telescope. For a circular aperture the pattern is known as the Airy disk (Airy, 1835), shown in Fig. 2.2. The size of the Airy pattern depends on the wavelength and aperture diameter:

$$\theta = \frac{1.22\lambda}{D}, \tag{2.8}$$

where θ is the angle defining the first light minimum, measured from the center, λ the wavelength, and D the telescope diameter.

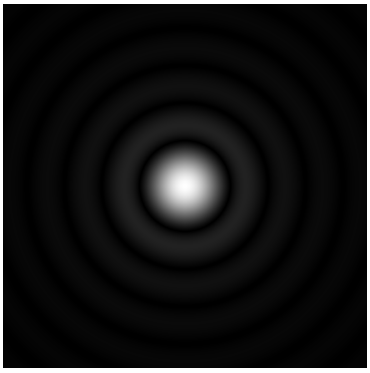


Figure 2.2: A simulated Airy disk.

The Airy disk is the result of an infinitely far point-source for an instrument limited by diffraction, that is, an instrument with perfect optics, without atmosphere. It defines the minimal size to which an optical system can focus light, i.e., its resolution. In practice, the diffraction pattern of large instruments is seldom a perfect Airy disk.

Firstly, in reflective telescopes the circular aperture is obstructed by the secondary mirror and its support. This results in a slightly softer diffraction pattern with long diffraction spikes.

Secondly, the optics are not perfect and may suffer from various aberrations:

- Defocus: when the image is not acquired at the focal position.
- Spherical aberrations: an aberration due to different striking light rays on spherical surfaces.
- Coma: an aberration due to the fact that incoming parallel rays striking the spherical surface with an angle are not all reflected to the same point.
- Astigmatism: an aberration where rays from two perpendicular planes are focused at different locations.

A convenient way to model these features is to work with the optical transfer function of the system in Fourier space. The PSF may be computed as the (inverse) Fourier transform of the autocorrelation of the complex entrance pupil, including the phase variations due to aberrations. This is how it is modeled in the SkyMaker astronomical image simulation software (Bertin, 2009),. Illustrations are presented in Fig. 2.3.

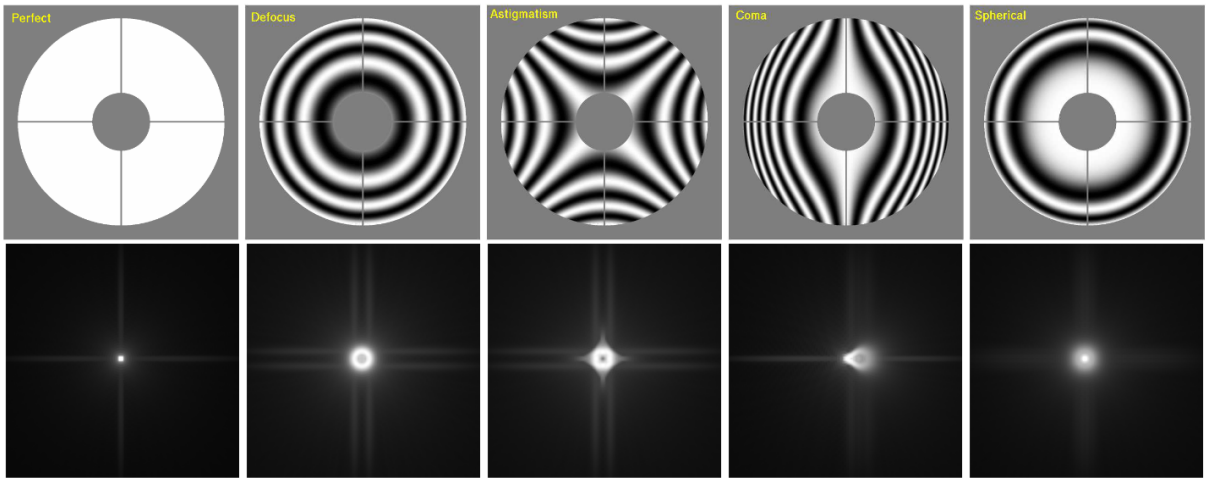


Figure 2.3: Real part of the pupil functions (top) and the corresponding PSFs (bottom) in the perfect case and in the presence of aberrations. Image credits: Bertin (2009).

Thirdly, other imperfections affecting the PSF include:

- Field curvature (also known as Petzval field curvature), which happens when the image surface is curved.
- Distortion, which happens when magnification varies across the focal plane.

2.4.3 Atmospheric turbulence

Apart from the instrument itself, which is generally well known, the other component of the PSF for ground-based observations originates from the turbulence of the atmosphere.

Because of air layers having different temperature and humidity levels, the refraction index is constantly changing throughout the atmosphere, distorting the path of light and thus the images. Wind is also playing a role by driving the turbulent pattern at different speeds and in different directions. Turbulence occurs at different heights: close to the ground (dome, surface boundary layer or due to ground convection), in the planetary boundary layer or associated with orographic disturbances (mountain ridges), and in the tropopause or above (Roddier, 1981).

The Kolmogorov model developed by Tatarskii (1961) and inspired by the studies of Kolmogorov (1941a,b) gives a mathematical description of the atmospheric turbulence which is found in reasonably good agreement with observations (Racine, 1996) for scale-lengths up to a few meters. It models the average width of a turbulence cell, r_0, also called Fried's seeing parameter, Fried's coherence length or Fried's r_0 as:

$$r_0 = 0.184\lambda^{6/5}(\cos\gamma)^{3/5}\left[\int_{path} dh(C_N(h))^2\right]^{-3/5}, \tag{2.9}$$

where:

- λ is the wavelength.
- γ is the angular distance from the zenith.
- h is the altitude.
- $C_N(h)^2$ is the refractive structure index coefficient, which models the average difference of refraction index between two points within the turbulent layer.

$C_N(h)^2$ can be measured using various methods that are not described here or modeled using mathematical functions which are often data measurement fits. A common model is the Hufnagel-Valley (Hufnagel, 1974; Valley and Wandzura, 1979). The coefficient is integrated across all the atmospheric layers in the optical path.

The optical transfer function (OTF) of atmospheric blurring in long exposures under a Kolmogorov model is written (Roddier, 1981):

$$\mathrm{OTF}(f) \propto \exp\left(-3.442\left(\frac{\lambda f}{r_0}\right)^{5/3}\right), \tag{2.10}$$

where $\|f\|$ is the angular frequency. This is the contribution of the atmospheric turbulence to the PSF. The larger the r_0, the better the conditions. It varies from about one centimeter in the worst sites in the blue band to tens of centimeters in the best sites at NIR wavelengths. Note that this is when pointing to zenith; r_0 increases with the zenith angle as the light path through the atmosphere lengthens.

In practice, turbulence has a major impact on the images from instruments with large apertures. The instantaneous effect for apertures $>> r_0$ is the presence of speckles characterized by an irregular distribution of bright stains and dark areas. This effect is due to the diffusion and the interference of the wave front occurring in turbulent cells throughout the atmosphere.

This can be really serious on very short exposures, as turbulence makes the "PSF" a stochastic process which may vary over very small angles (for high altitude layers). Yet, in our observation regime, i.e., wide-field optical and NIR images with long exposure times, numerous speckles stack up during the exposure so that the PSF results in a large blurred stain, called *seeing disk*. The full width at half maximum (FWHM) of the seeing disk, or simply *seeing*, is often used as an empirical measure of the quality of the atmosphere. The higher the amount of atmospheric turbulence, the larger the FWHM. It is related to r_0 with:

$$\mathrm{FWHM} \approx \frac{0.98\lambda}{r_0}. \tag{2.11}$$

Seeing is generally the dominant contributor to the spread of the PSF as shown in Fig. 2.4.

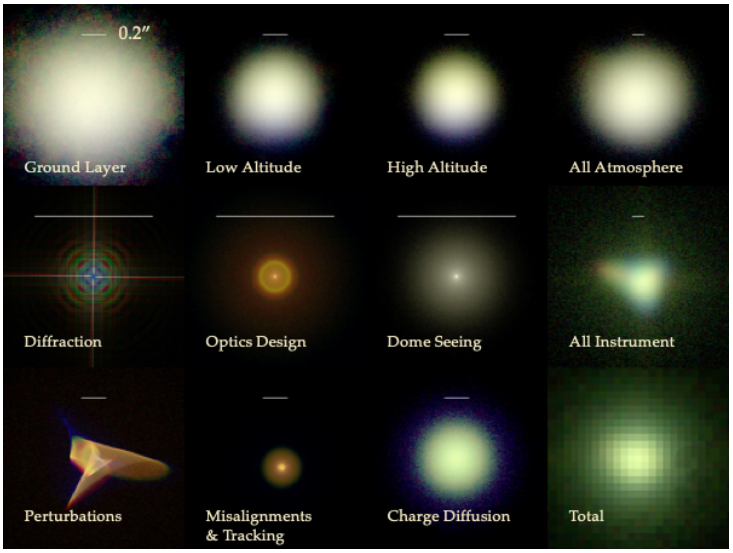


Figure 2.4: Illustration of the various contributions to the PSF that are taken into account in the PhoSim image simulation software (Peterson et al., 2015), including atmospheric turbulence. Image credits: Peterson et al. (2015).

2.4.4 Aureole

Diffusion effects dominate the PSF beyond several FWHMs from the center, causing a faint halo called aureole. Even though the exact origin of the aureole is not well understood, it is believed to originate from a combination of instrumental and atmospheric light scattering (Racine, 1996). It generally follows a power law (King, 1971; Racine, 1996).

2.4.5 Pixel response

Finally, the PSF must also account for the intra-pixel response function, i.e., the variation of detector sensitivity below the scale of a pixel. It is particularly significant in critically undersampled instruments such as Euclid because the actual optical PSF is concentrated over a few pixels (Shapiro et al., 2018). A perfect pixel response function would correspond to a door function of the size of the detector's pixels. Because of charge diffusion, the pixel response function often shows a tail that extends beyond the pixel footprint.

2.5 Electronic detectors

After traveling through space and the atmosphere and being focused by the telescope, photons emitted by astronomical objects are transformed into a numerical image by electronic detectors at the focal plane. The electronic detector is either a CCD (charge-coupled device), or a CMOS (complementary metal oxide semi-conductor) device. Both types of detectors are arrays of electric potential wells, that define the pixels, where light is converted into a voltage. Both use the photoelectric effect: incident photons hit the substrate, commonly silicium, where they excite electrons that enter the conduction band. These electrons are then collected in each well and read. In CCDs charges are transferred from well to well to an output amplifier whereas in CMOS this process is done at each pixel independently. Fig. 2.5 shows an illustration of both reading techniques.

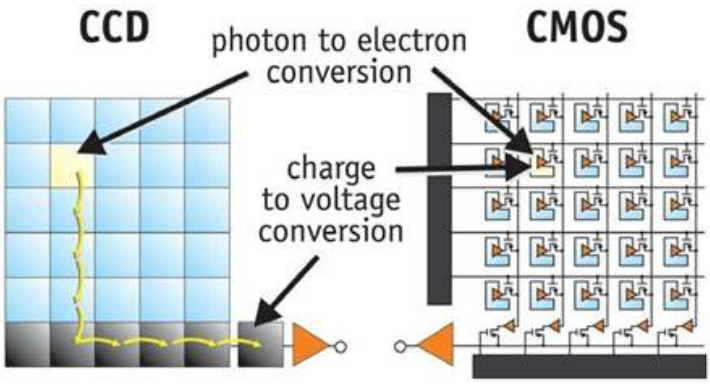


Figure 2.5: Comparison of CCD and CMOS detector readout strategies. Image credit: Stefano Meroli (CERN).

Both types of detectors introduce additional "signals":

- Bias: this is the offset signal emitted by the electronics which does not depend on the exposure time.
- Dark offset: this is the signal due to the dark current, i.e., to the spontaneous and random generation of electrons in the detector even when no photons are collected. This process is of thermal origin and amplifies with detector temperature.
- Readout noise: this is the Gaussian noise generated by the on-chip amplifiers and the sampling process.

The continuous voltage signal generated by the accumulated charges is transformed into a discrete digital signal by the analog-to-digital converter. This quantification process introduces a quantization error corresponding to the rounding error between the analog voltage and the digitized values. Astronomical images are generally stored using single-precision floating-points in FITS format (Wells et al., 1981), standing for Flexible Image Transport System.

2.6 Image calibrations

At a given pixel i, detector measurements can be written as:

$$y_i = a_i x_i + b_i + n_i, \tag{2.12}$$

where:

- y_i is the observed value of the pixel.
- x_i is the "true" value of the pixel.
- a_i is a multiplicative gain factor which incorporates the quantum efficiency of the pixel as well as local attenuation due to optical vignetting and dust or spots in the optical path.
- b_i is the additive term corresponding to the bias of the detector and the mean dark current offset.
- n_i is a random variable introduced by the Gaussian readout noise.

For the whole image:

$$\boldsymbol{y} = \boldsymbol{a} \odot \boldsymbol{x} + \boldsymbol{b} + \boldsymbol{n}, \tag{2.13}$$

where $\odot$ denotes the point-wise product.

The usual correction procedure is:

$$\boldsymbol{y'} = (m(\boldsymbol{y} - \boldsymbol{d})) \oslash (\boldsymbol{f} - \boldsymbol{d}), \tag{2.14}$$

where:

- $\oslash$ denotes the point-wise division.
- $\boldsymbol{y'}$ is the corrected or reduced image.
- $\boldsymbol{y}$ is the raw image as defined in Eq. 2.13.
- $\boldsymbol{d}$ is a dark frame: an exposure of the same duration as $\boldsymbol{y}$ with the shutter closed. It is subtracted from $\boldsymbol{y}$ to correct for the bias and dark offset.
- $\boldsymbol{f}$ is a flat field: an exposure of a angularly uniform light. It is bias-subtracted and normalized by m.
- m is an arbitrary factor, generally taken as the median of $\boldsymbol{f} - \boldsymbol{d}$.

The image $\boldsymbol{y'}$ is said to be bias and flat field corrected. Note that raw $\boldsymbol{d}$ and $\boldsymbol{f}$ exposures are also subject to noise themselves. A common procedure to reduce noise is therefore to average several dark and flat-field exposures to generate "master" dark and flat frames. The corrected images $\boldsymbol{y'}$ are usually called "science" images, as opposed to the raw images $\boldsymbol{y}$. From now on, the images discussed in this work are considered to be science images. In such images, only the contribution of the sky background and sources remains.

2.7 Sources in astronomical images

The two main types of sources of interest for detection are stars (point-sources) and galaxies (extended sources).

2.7.1 Stars

Stars (and quasars) are point-sources at the scale of a fraction of an arcsecond (the typical angular pixel size for optical and near-infrared wide-field cameras on large telescopes). As stated in Section 2.1, in our simplified and generic image model, we describe the resulting image as the convolution of true star signals with the PSF of the instrument, (ignoring the two sources of noise). A basic example is shown in Fig. 2.6.

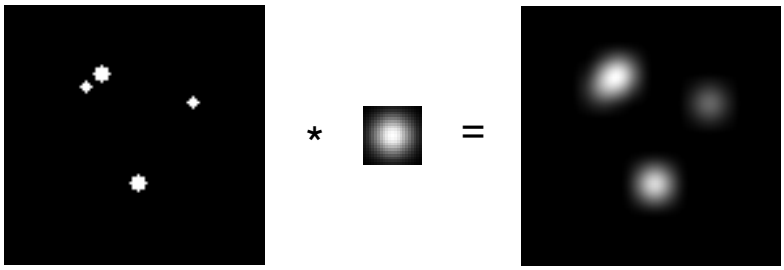


Figure 2.6: Simplified illustration of the image of stellar sources obtained with a telescope. Left: true signal. Middle: PSF of the instrument. Right: result image. The result image is the convolution of the true signal with the PSF. I do not include sky background, noise and pixel sampling in this representation. Note how the two close sources in the top left are blended in the result image.

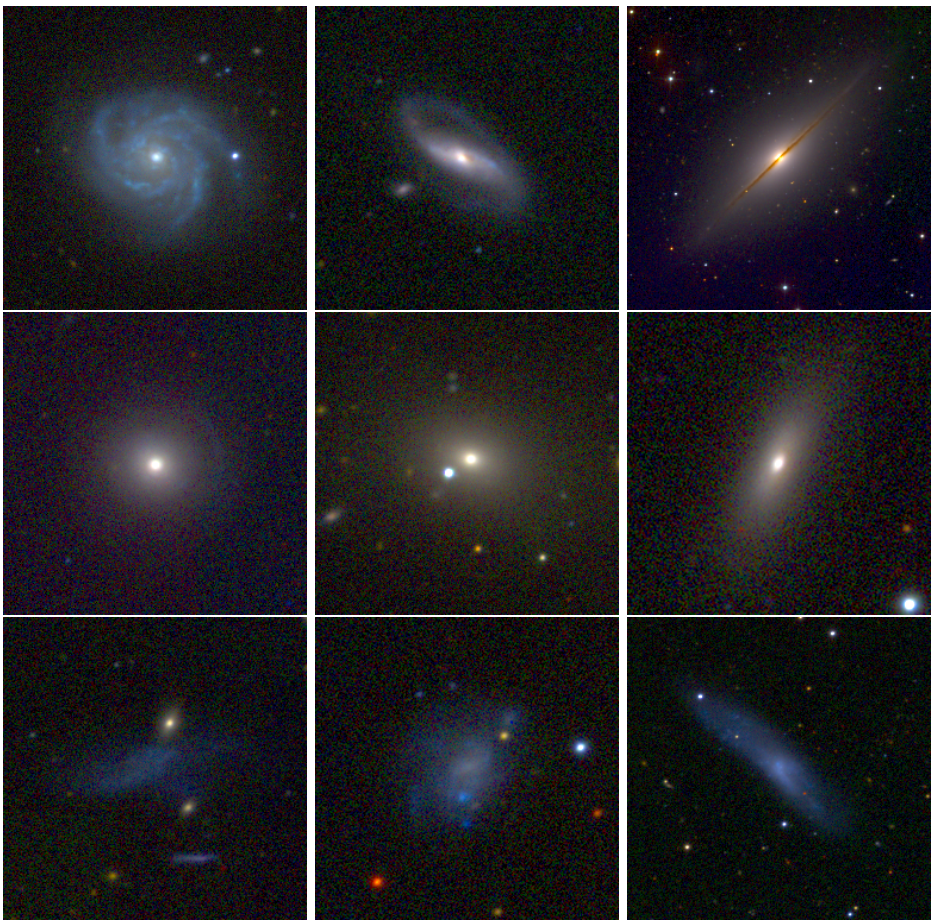


Figure 2.7: Examples of the 3 main morphological types of galaxies. Top: spiral. Middle: elliptical. Bottom: irregular. Images credits: EFIGI project (Baillard et al., 2011).

2.7.2 Galaxies

Galaxies have a more complex appearance. They show up as extended sources that can have various shapes: elliptical, spiral or irregular as shown in Fig. 2.7. Some galaxies can exhibit several components, be asymmetric or non convex.

Even within each morphological type a galaxy may look very different depending on the point of view and on the peculiarities within each type. For example, spiral galaxies can exhibit different arm patterns, have different bulge sizes, and can be seen under various orientations (face-on, on the edge, etc); elliptical galaxies may appear more or less flattened, etc.

Within our image model, we obviously describe galaxy images as the convolution of the true light distribution with the PSF. However, except for the closest galaxies, they cannot be resolved in stars, at least from the ground.

2.8 Conclusion

In the scope of designing a robust and universal source detector for optical and NIR wide-field images, we have defined a simplified yet generic image model. Astronomical images can be modeled as the result of the convolution of the true signal of interest with the PSF of the instrument plus an additional noise. The PSF depends on the instrument and generally varies within the field of view. It is often dominated by atmospheric turbulence which varies with time, for ground-based instruments. Sources lie on top of a sky background, which is often made uneven from contamination by stray light and extended nebulae. In practice, most faint sources look like small blurs, more or less diffuse, with no clear boundaries. Complicating this, background noise (Poisson noise from the sky plus readout noise) makes the object even more elusive. Finally, source blending in dense regions makes it challenging to detect individual objects, let alone individual contributions to the total light.

Because of all these issues, identifying sources in images is not an easy task, and raises questions such as: given the nature of astronomical images, what is the best feature for identifying a source ? Existing solutions rely on at least one of the following source features:

- Source peak: a source can be identified by the mode of the light distribution. This can be problematic regarding some extended sources that do not exhibit a unique intensity peak.
- Source centroid: a source can be identified by its central part, generally defined as the barycenter of the light distribution. This may be inappropriate for strongly skewed source profiles.
- Source pixels: a source can be identified by a pixel mask. The mask may correspond to, e.g., an area enclosing some fraction of the estimated source flux, or a set of connected pixels with values exceeding some threshold above the sky level.

With a proper model of astronomical images in hand, let me now review the existing solutions to source detection.

Chapter 3

Existing solutions to source detection

In this chapter, I review the state-of-the-art of source detection techniques from the past decades[1].

One may distinguish two main types of approaches: basic detection algorithms, consisting of background estimation, filtering, local peak search or thresholding, and multiscale approaches, mainly based on wavelet transforms. I review all these techniques in the following sections, indulging in a few digressions on related methods or topics where applicable.

Note that some of the cited methods are also applied to X-ray or radio images even if this project is more about optical and NIR data.

3.1 Basic detection algorithms

Basic detection algorithms involve several of the following steps:

- Sky background estimation and subtraction or other preprocessing,
- Matched filter.
- Local peak search.
- Thresholding.
- Source fitting.
- Deblending.

3.1.1 Preprocessing and sky background estimation

In order to be able to detect the faintest objects and/or to derive accurate photometry, i.e., estimate source light intensities (fluxes), it is essential to have a precise measurement of the contribution of the sky background. Theoretically, there should be one background map per source, each describing how the image would appear without the source. However, no methods have been developed to achieve it. This is why most approaches compute a single background map per image.

To retrieve the sky background level from images, most methods rely on the assumption that if the image is made mostly of "background" pixels, the histogram mode of the image can provide an appropriate estimate of the sky background level. Therefore, the aim of the majority of sky

[1] Bertin (2001) and Masias et al. (2012) are valuable resources reviewing source detection methods until the last decade.

background estimation methods is to fit models to the image histogram or to design algorithms to estimate its mode.

For instance, Bijaoui (1980) uses a Bayesian model to fit the image histogram and derive several sky background parameters such as the sky background level from this model. Even though it is used in several other studies (Irwin, 1985; Le Fevre et al., 1986; Slezak et al., 1988), it proved too computational expensive for most applications at the time.

Thus, simpler and faster methods based on mode estimations have been preferred in practice. In addition, since the sky background may vary across the field, most of these methods make local estimates of the sky background level. Thereby, the entire field is divided into smaller images and a sky background value is estimated within each of these images. A full sky background map can then be built by interpolating these values, using, e.g., bicubic spline interpolation. Sometimes, the local values are also median or Gaussian filtered prior to interpolation in order to mitigate local overestimates.

Some of the early methods processing this way relied on the local mean (Herzog and Illingworth, 1977; Kron, 1980). However, since the mean is very sensitive to outliers, the median is often preferred, e.g., in Damiani et al. (1997) after smoothing with a Gaussian filter, in Mopex (Makovoz and Marleau, 2005), or in Lang et al. (2010), the latter estimating the standard deviation of the sky background by picking random pixel pairs.

Other methods estimate the sky background level after pre-detection of sources, like Buonanno et al. (1983a), Daophot (Stetson, 1987) and Yee (1991) that estimate the sky background value as the mode of an annular region around the stars. This method is also used in Vikhlinin et al. (1995) and Szalay et al. (1999) to compute the sky background level in X-ray images and for faint object detection, respectively.

A better compromise between accuracy and speed is reached with iterative estimations of the mode of the image histogram. Differences between mean, median and mode for sky background estimation are illustrated in Fig. 3.1. We can clearly see that the mean and median are poor approximations of the mode in crowded fields. Moreover, the mode may not be a good estimator of the sky background level in this regime.

In order to make more robust estimations of the sky background level, a lot of methods use algorithms to discard some image pixels. For example, Lasker et al. (1990) simply take the mean of empirically clipped histogram values. On the other hand, SExtractor (Bertin and Arnouts, 1996) uses an automatic procedure called k-σ clipping. This procedure consists of discarding pixels which values are above the sum of the mean and k-σ. This process is iterated until no more pixels can be discarded and the remaining pixels are used to estimate the sky background level, using the following criterion: if σ changed less than 20% during the k-σ clipping procedure, the field is considered uncrowded and the retained sky background value is the mean of the clipped histogram. Otherwise, the sky background value b is given by

$$b = 2.5 \times \text{median} - 1.5 \times \text{mean}, \tag{3.1}$$

a modified version of Person's rule (Pearson, 1895), which normally uses 3 and 2 as approximation coefficients, as in, e.g., Kendall and Stuart (1977) and later versions of Daophot. The k-σ clipping procedure has also been used in Lazzati et al. (1999) for X-ray images and in Perret et al. (2009).

Finally, some alternative histogram model fitting are still used in Szalay et al. (1999) and Hopkins et al. (2002) with radio images. A Gaussian of parameter (μ, σ) is fitted to the histogram. The original image is then normalized by subtracting μ and dividing by σ.

After having estimated and subtracted the sky background from the image, most "classical" methods rely on the matched filter.

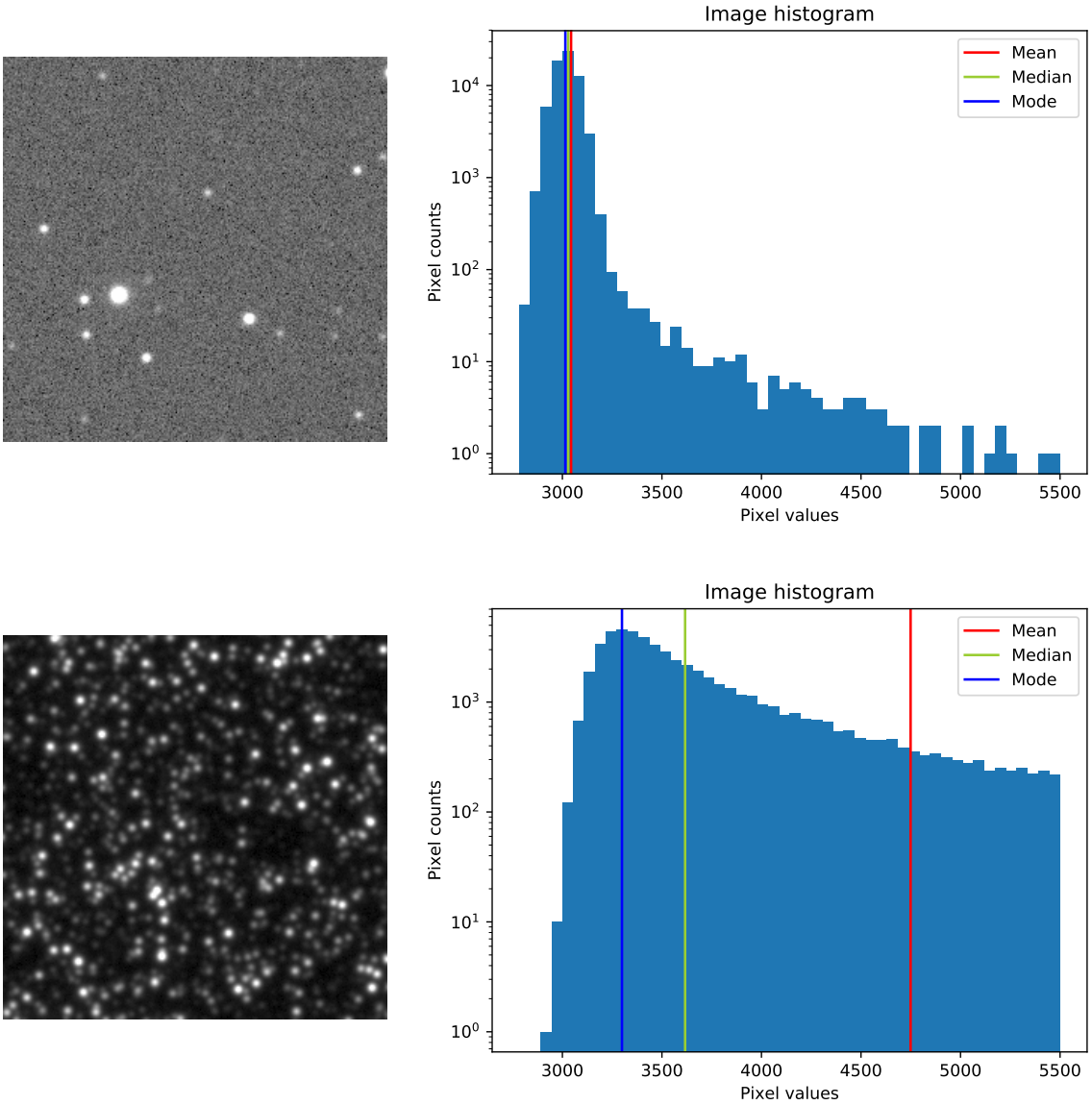


Figure 3.1: Illustration of different sky background approximations. Top: a low density stellar field and its histogram. Bottom: a high density stellar field and its histogram. Mode, median and mean are represented in each histogram. In the low density stellar field, the median and the mean can make good approximations of the mode but it is not the case in the high density stellar field.

3.1.2 Matched filter

The matched filter (Woodward, 1953, 2014; Turin, 1960) is used to enhance the contrast of known patterns in noisy images. It consists in correlating the input with a template to detect the presence of the pattern in the input. In the one-dimensional continuous case, cross-correlation is defined as:

$$c(t) = (f * h)(t) = \int_{-\infty}^{\infty} f(\tau + t) \cdot h^*(\tau) d\tau = \int_{-\infty}^{\infty} f(\tau) \cdot h^*(\tau - t) d\tau, \tag{3.2}$$

where h^* denotes the conjugate of h.

When searching for a correlation template h_m maximizing the signal-to-noise ratio of a signal s in the presence of stationary noise (in the least square sense), one finds:

$$H_m = \frac{S^*}{\mathcal{P}}, \tag{3.3}$$

where $\mathcal{P}$ is the noise power spectrum and H_m and S are the Fourier transform of h_m and s, respectively. In the case of a point-source over white noise, the power spectrum $\mathcal{P}$ is constant and the template is simply the instrument's PSF. As the PSF is often symmetric, applying the matched filter consists in convolving the input with the PSF (a simple point-wise product in Fourier space). An illustration of the 1D matched filter is shown in Fig. 3.2.

The matched filter is optimal only in the presence of (wide-sense) stationary noise, i.e., a noise that keeps the same (2nd order) statistical properties across the image. In our case, this condition is generally verified over vast portions of wide-field optical and NIR images with long exposure times, as already stated in Section 2.3.

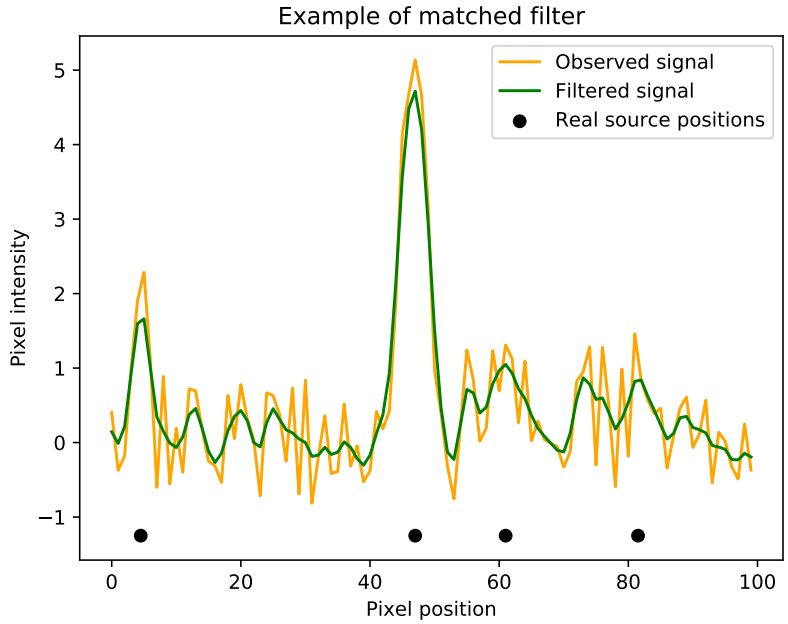


Figure 3.2: Illustration of the 1D matched filter. The sky background value is assumed to be zero. The observed signal is comprised of sources with Gaussian profiles with unit standard deviation on top of white Gaussian noise with $\sigma_{\mathrm{SKY}} = 0.5$. Matched filtering is carried out by convolving the signal with a Gaussian kernel with unit standard deviation.

Yet, a strong limitation of the matched filter is that it is ideal only to detect isolated signals. It is thus well suited to identify isolated sources, but it is limited in crowded fields where sources overlap because the noise is not stationary anymore.

Confusion noise regime: An extreme case of crowding is the confusion noise regime. It happens in very dense fields when the sky *background* is dominated by the faintest unresolved sources. Illustrations of this regime are shown in Fig. 3.3. In such a regime, the noise cannot be approximated as white noise. Instead, it is a Poisson noise convolved with the local PSF so that its power spectrum $\mathcal{P}$ is no longer flat. Then the optimal filter is in fact the direct deconvolution of the image with the PSF and only the brightest peaks are detected in the image.

One of the first use of the matched filter for automatic astronomical source detection is done in Irwin (1985), suggesting the seeing function as template, which can be estimated by "directly averaging suitable stellar profiles or by an analytic model fit to these profiles". DAOPHOT (Stetson, 1987) and Hopkins et al. (2002) filter the image with a Gaussian kernel while a Gaussian fit of the PSF is done in Slezak et al. (1988) as template filter. Mighell (1989a, 1999) suggests smoothing high frequencies, that are likely to be noise, by using a low pass filter template. The matched filter is extensively used, e.g., in Vikhlinin et al. (1995) with X-ray images or in SExtractor (Bertin and Arnouts, 1996). The matched filter is still the basis for contemporary source detection algorithms (Maddox and Dunne, 2020).

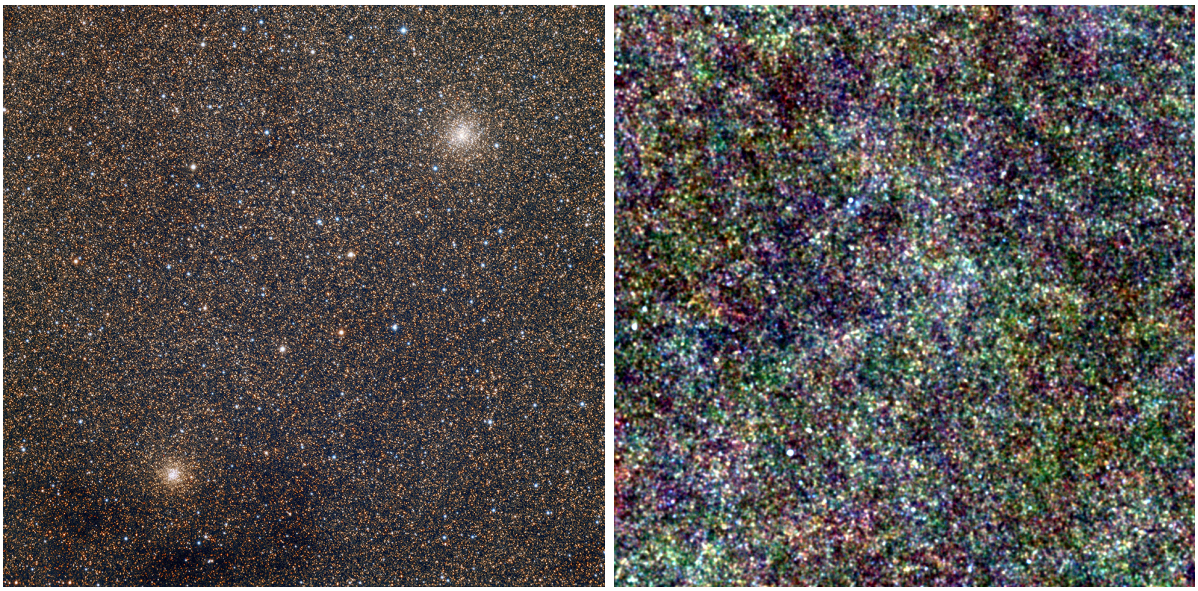


Figure 3.3: Illustrations of the confusion noise regime. Left: Baade's window. Right: Lockman's hole. Image credits: Adam Block/Mount Lemmon SkyCenter/University of Arizona and SPIRE instrument (Griffin et al., 2010).

Variations to the basic matched filter have been developed. A non-linear matched filter (Makovoz, 2005), less computationally extensive, is implemented in MOPEX (Makovoz and Marleau, 2005). Matched filters have also been used in multi-channel images. Melin et al. (2006) filter each channel independently and create a single filtered image. Herranz and Sanz (2008); Herranz et al. (2009) use $N_c \times N_c$ filters, called matrix filters, to compute one filtered image per channel using all channels.

Once the image is filtered and source contrast is enhanced, two main methods are used to do the detection: local peak search and thresholding.

3.1.3 Local peak search

Local peak search, or maximum search, consists of finding peak pixels, i.e., local maxima.

$$I_{lp}(p) = \begin{cases} 1 & \text{if } \forall q \in \mathcal{N}_p, I(p) \geq I(q) \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

where $\mathcal{N}_p$ is the set of pixels neighboring p. An illustration of local peak search is shown in Fig. 3.4 below, reusing the example of Fig. 3.2.

For instance, Herzog and Illingworth (1977) and (Newell and O'Neil, 1977) define peaks using two criteria: the pixel value must be greater than a constant above the sky background

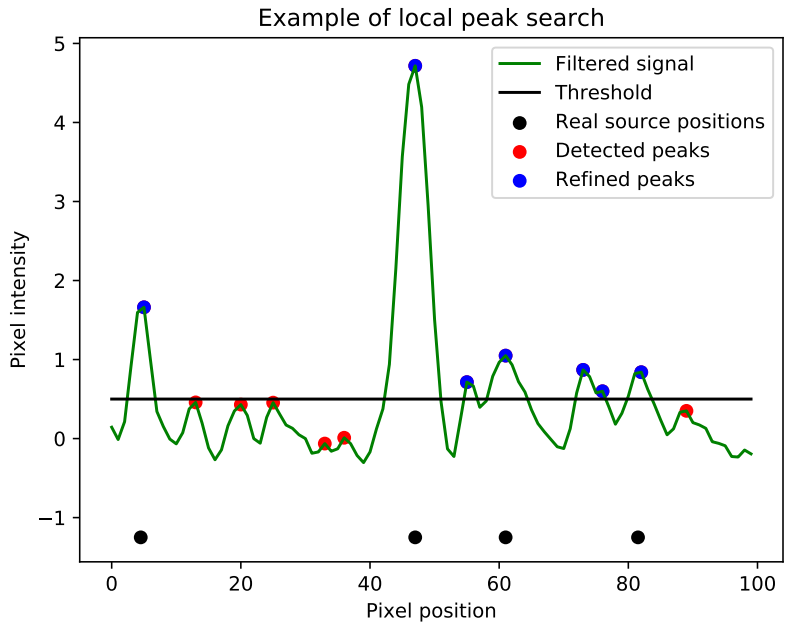


Figure 3.4: Illustration of a local 1D peak search. Peak pixels are defined as having values higher than those of the two immediate neighbors. Adding a threshold condition may help refine the selection of detected peaks.

and its eight closest neighbors. Kron (1980) and Yee (1991) use the latter criterion and then verify that the average of the given pixel and its eight neighbors is greater than some fraction of the sky background value. Buonanno et al. (1983a) identify pixel values higher than that of their neighbors, retaining only those above some value and using a contiguity criterion to avoid multiple detections of the same object. DAOPHOT (Stetson, 1987) checks for pixel value above a predefined threshold and above those of neighbors closer than a distance d, where d is estimated using a user-supplied FWHM. Lang et al. (2010) retain pixels above 8σ of the sky background and pick those that have higher values than the neighbors. They trim the smaller peaks by looking for those that are joined to smaller peaks by saddle points within 3σ of the larger peak (or 1% of the larger peak's value, whichever is greater). Other studies supporting local peak search include Mighell (1989a,b); Vikhlinin et al. (1995); Mighell (1999).

The main weakness of local peak search comes from not being suited to extended objects with diffuse patterns and no clear maximum.

3.1.4 Thresholding

The thresholding operation turns an image into a binary image, also called a segmentation map. It operates on background-subtracted, matched filtered pixels. Each pixel is assigned a binary value in the segmentation map depending on whether it sits above or below the threshold:

$$I_t(p) = \begin{cases} 1 & \text{if } I(p) > t \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

where t is the threshold value. The two resulting classes of pixels usually end up being objects and background. An illustration of thresholding is shown in Fig. 3.5.

The most critical point when using thresholding is to define an appropriate threshold: too low and noise peaks may trigger false detections, too high and the faintest sources may be missed.

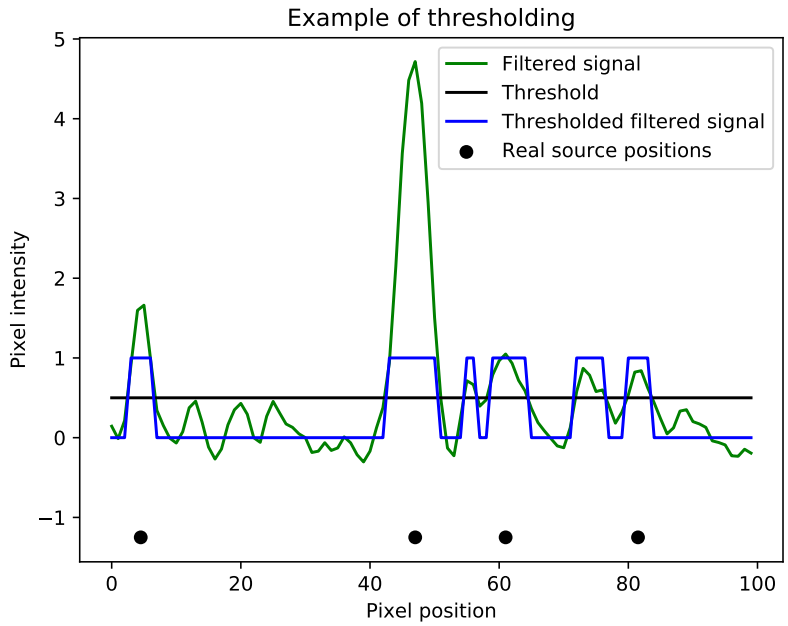


Figure 3.5: Illustration of 1D thresholding. Values above the threshold are set to 1 while values below are set to 0.

Irwin (1985) simulates sources with Gaussian distribution and random noise to estimate the completeness and contamination levels for different thresholds.

Le Fevre et al. (1986) use the standard deviation of a Gaussian model adjusted to the histogram of the sky background to define the threshold. The threshold is set to 1.5σ by default, which is quite low but false positives are prioritized over false negatives in their study. Slezak et al. (1988) also use the standard deviation of a Gaussian fit, but the latter is made on a 8 maximum neighbors histogram. The threshold is set at 3.8σ.

Lasker et al. (1990) simply threshold at 125% of the sky background value. This threshold is increased for crowded fields.

Szalay et al. (1999) use a threshold to retain pixels that are unlikely to be background assuming that the sky background is Gaussian distributed. It optimally happens at the intersection between a sky background Gaussian histogram and the image histogram (Fukunaga, 1990). In practice, it corresponds to 2.43σ of the sky background in their application. Hopkins et al. (2002) choose a similar criterion to define the threshold in radio images.

Detection thresholds, performance metrics and ROC curves: Various detection thresholds may be used, resulting in different performance trade-offs. Detectors can be seen as 2-class classifiers and their performance is usually assessed by counting how many true and false objects are detected. These are referred as true positives and false positives, respectively. After thresholding, pixels above the threshold are considered as predicted positive, i.e., objects, while pixels below are considered as predicted negative, i.e., background. If one knows the ground truth of each pixel, i.e., if each pixel is actually object or background, then one can count the number of true positive (TP), false positive (FP), true negative (TN) or false negative (FN) pixels as shown in Table 3.1.

Common performance metrics can be derived from TP, FP, TN, and FN:

		Actual class	
		Positive/Object	Negative/Background
Predicted class	Positive/Object	TP	FP
	Negative/Background	FN	TN

Table 3.1: Basic performance metrics (see text).

$$\text{True positive rate: } \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{TP}}{\text{P}} \tag{3.6}$$

$$\text{False positive rate: } \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} = \frac{\text{FP}}{\text{N}} \tag{3.7}$$

$$\text{Purity: } \text{PUR} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.8}$$

$$\text{Accuracy: } \text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}}, \tag{3.9}$$

where P and N denote the number of actual positives and negatives, respectively. The true-positive rate is the ratio of true positives to all the actual positives. The closer to 1, the better the classifier or detector. It is also referred as sensitivity or recall. The false-positive rate is the ratio of false positives to all the actual negatives: the closer to 0, the better. Purity is the ratio of true positives to all the predicted positives; the closer to 1, the better. It is also referred as precision. Accuracy is a global performance measure. It gives a quick idea of the classifier performance; the closer to 1, the better.

A common practice is to represent the classifier performance using a ROC (Receiver Operating Characteristic) curve. It represents the TPR versus the FPR for various detection thresholds. Examples of ROC curves are given in Fig. 3.6.

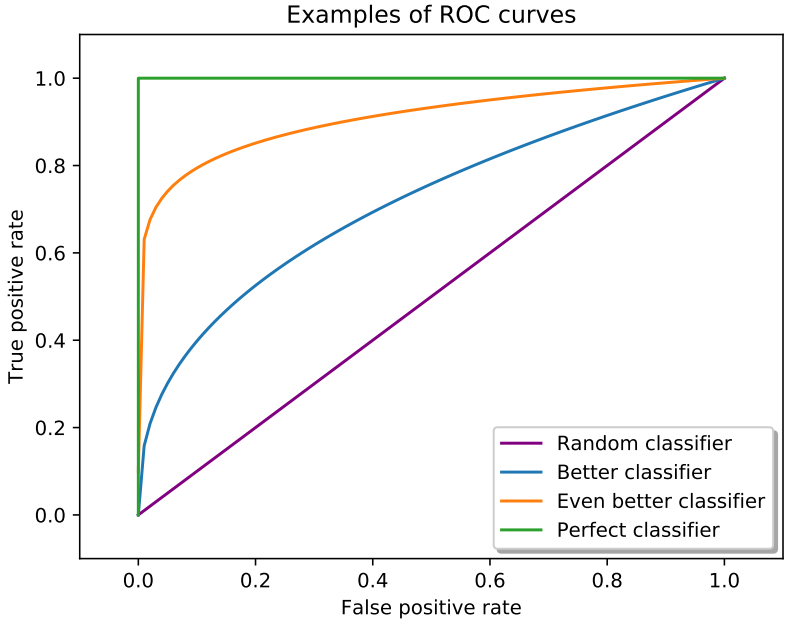


Figure 3.6: Examples of ROC curves. The more a curve bends towards the top left, the better the classifier performance is. Each point in a ROC curve corresponds to a (FPR, TPR) couple at a given probability threshold in $[0, 1]$.

The more a ROC curve bends towards the top left, the better the classifier is because it means that the TPR and FPR get closer to 1 and 0, respectively. Another performance metric derived directly from the ROC curve is the AUC, standing for area under curve, which is simply the area under the ROC curve. The closer it is to 1, the better the classifier or detector is. All these performance metrics can be used to assess the performance of a detector or to compare several detectors.

The segmentation map obtained after thresholding gives the list of pixels considered as objects and those considered as background. However, it is far more convenient to assess the performance of a detector at the object level rather than at the pixel level. For that, pixels need to be assigned to individual objects. This can be done using a connected component analysis (Rosenfeld and Pfaltz, 1966). Pixels are grouped according to a connectivity criterion. The most widely used criteria are 4-neighbor connectivity and 8-neighbor connectivity, using 3×3 cross and square shapes, respectively. It is then more suitable to compute the performance metrics mentioned above at the object level.

While isolated objects can be properly identified through connected component analysis, blended objects remain identified as single objects at this point. This is why deblending procedures are also applied to separate multiple objects detected as single ones.

3.1.5 Deblending procedures and source fitting

Deblending procedures can take place before or during source fitting. Source fitting aims to find the light profiles that best fit the sources.

Newell and O'Neil (1977) and Herzog and Illingworth (1977) were among the first to design a deblending procedure, called DOG (Data Over Gradient), after local peak search. The idea behind the DOG procedure comes from the observation that a Gaussian distribution divided by its gradient is the inverse function, which provides much sharper maxima. Assuming that source profiles are Gaussian, the transformation is applied to the image and a second local peak search is done.

Other methods address deblending via multiple source fitting. For instance, Buonanno et al. (1983a) define an "action area" radius for each source during the local peak search. A deblending procedure is then run if several sources have overlapping action areas. The procedure consists in fitting multiple components (Fraser and Suzuki, 1966), later extended to two dimensions. Components are analytical approximations of the PSF (King, 1971). The components are circular Gaussian functions in Buonanno et al. (1979), and Moffat profiles in Buonanno et al. (1983b); Moffat functions are known to be good approximations of ground-based PSFs (Moffat, 1969). Moffat profiles are also used in Mighell (1989a,b) while Penny and Dickens (1986) is another method using Gaussian profiles. Lorentzians, which are particular Moffat profiles, have also been used, e.g., in Franz (1973), and Penny (1979). DAOPHOT (Stetson, 1987) applies a similar strategy for deblending, but using a more sophisticated PSF model derived from a selection of stars in the field.

Le Fevre et al. (1986) propose an algorithm to detect blends that will later go through visual inspection: it starts from the maximum of each detected component during thresholding and iteratively extends radially in each direction to fit an ellipse. If a good fit happens before reaching an intensity of 2σ of the estimated sky background, then the source is considered multiple and is stored for visual inspection, where it is manually delimited by an ellipse. Lasker et al. (1990) apply a local peak search on the thresholded connected pixels to find blended sources. After this, they correlate the detected sources with predefined clean source profiles from a library. Slezak et al. (1988) directly fit ellipses to the thresholded pixels and tries to tackle blended objects during a star/galaxy separation procedure.

Another method commonly used by several authors is multi-thresholding. Indeed, blended sources detected as a single object at a given threshold may be separable at a higher threshold. The multiple thresholds used are usually logarithmically spaced. For instance, Irwin (1985) uses logarithmically spaced thresholds every quarter magnitude interval. The potential deblended sources are then fitted with circular Gaussian profiles. SExtractor (Bertin and Arnouts, 1996) also uses multi-thresholding followed by bivariate Gaussian fits, while Mopex (Makovoz and Marleau, 2005) applies a multi-component fit based on the number of deblended sources by multi-thresholding. If the fitting quality is below a user specified threshold, the number of component is incremented to check if it fits better with one more source in the blend. If it does not, it reverts to the initial number of sources in the blend. An illustration of multi-thresholding is shown in Fig. 3.7.

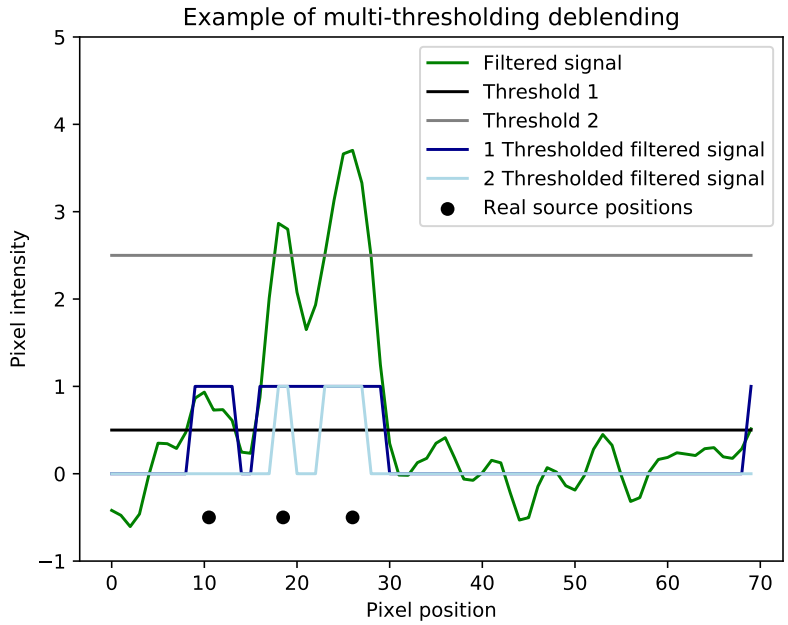


Figure 3.7: Illustration of deblending with multi-thresholding. The multiple source is detected as a single object with threshold 1 but it is well detected as a multiple object at higher threshold 2.

Even recently, multi-thresholding is still used for deblending purposes (Zheng et al., 2015). Yet, some new techniques for deblending have been developed. For example, the Lupton algorithm[2] (unpublished) detects the individual sources of a blend by finding peaks. Then, it models the blend as the linear combination of each detected source and fits a template for each source using a symmetry hypothesis. Recently developed methods also include Muscadet (Joseph et al., 2016) that uses a morpho spectral component analysis based on morphological dictionaries and SCARLET (Melchior et al., 2018) that uses constrained matrix factorization. Both assume that a blended image is the linear combination of the contributions of each individual source. Their aim is then to recover the individual source images.

At this point, it is important to discuss the two different levels of deblending, changing the meaning of the word when employed in the literature. Indeed, there is the deblending at the detection level and the deblending at the measurement level. Those imply two different tasks, so that the word *deblending* in the literature can implicitely refer to one or the other. In particular,

[2]https://www.astro.princeton.edu/ rhl/photomisc/deblender.pdf

in the most recent literature, deblending almost always refers to the measure and the deblending task consists of recovering the individual fluxes of the blended sources. On the other hand, deblending at the detection level consists of detecting that some image is made of a blend of several sources. In this work, we focus on tackling deblending at the detection level, and there are no recent works tackling deblending at this level to our knowledge.

3.1.6 Discussion

One strong limitation of all the previous techniques is that they remain quite empirical and heuristic based, i.e., they use practical methods that are not guaranteed to be optimal, such as the background estimation techniques, thresholding or the deblending routines. Each one must apply small changes and variations to each method to make it work better with their application. The size of the local area of background estimation, the kernel to use in the matched filter, the setting of the detection threshold, etc. Each step in the whole processing depends on parameters that need to be tuned to work well with given data in practice. Especially, extensive tuning is necessary when processing higher source density regions or to detect particular objects such as low surface brightness galaxies. Thus, one must constantly inject prior knowledge, tune parameters to make it work on its data and make compromises between the types of objects to detect.

Among the tunable parameters, a non exhaustive list counts a parameter to ignore a certain number of the higher pixels in the median filter evaluation of the sky background in Makovoz and Marleau (2005), the sharpness and roundness criteria in DAOPHOT (Stetson, 1987) to avoid detecting cosmic rays or bleeding saturations, the minimal size of objects to consider as true detections after thresholding in SEXTRACTOR (Bertin and Arnouts, 1996), etc. Each pipeline has a lot of tuning routines like those to adapt to its data.

The whole reduction from an image to a source catalog relies on the chain of all these processes, where any parameter change at one stage can have consequences later in the pipeline. Two pipeline examples are given in Fig. 3.8.

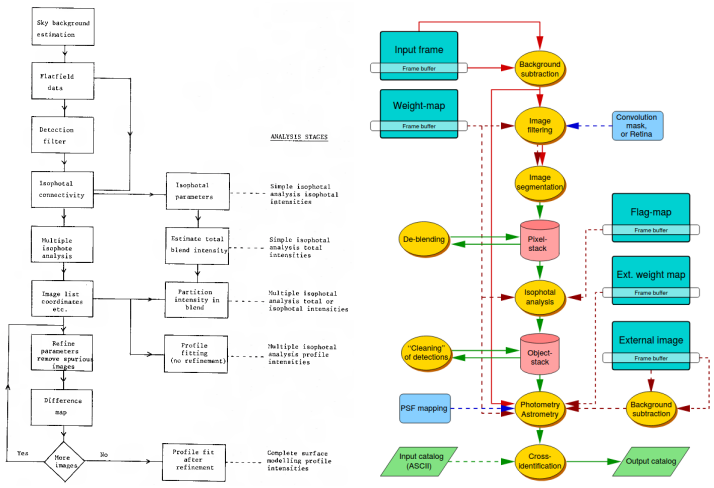


Figure 3.8: Two examples of source detection pipelines. Left: Irwin (1985). Right: SEXTRACTOR (Bertin and Arnouts, 1996). The purpose is just to show the complexity of the pipelines. Each block is heuristic based and the final detection results rely on each of the processing blocks. Images credits: Irwin (1985) and AstrOmatic SEXTRACTOR documentation: https://www.astromatic.net/software/sextractor.

The quality of the detection also depends on the sky background and PSF estimations, which both remain challenging and open problems.

3.2 Methods using mathematical morphology

Mathematical morphology (Serra, 1982, 1988) is a theory based on set theory and topology, created by J. Serra and G. Matheron in 1964 (Matheron and Serra, 2002) for analysing geometrical structures.

3.2.1 Mathematical morphology

Mathematical morphology (MM) was first designed around binary images, where 1 may be considered as a foreground and 0 as background. MM operators are based on the presence of a pattern, called the structuring element. The two main morphology operations are erosion and dilation, which reduce and increase the footprint of foreground pixels, respectively. Formally, the operations are defined as:

$$\text{Erosion: } \epsilon_S(I) = I \ominus S = \{p \in I | S_p \subseteq I\} \tag{3.10}$$

$$\text{Dilation: } \delta_S(I) = I \oplus S = \{p \in I | S_p \cap I \neq \emptyset\}, \tag{3.11}$$

where I is the input image and S the structuring element. The structuring element is moved across all positions in the image for both operations. For the erosion, the new pixel p value is 1 if S_p, the structuring element at this position, is included in the input foreground, and 0 otherwise. For the dilation, the new pixel p value is 1 if S_p has at least one pixel in common with the input foreground, and 0 otherwise. An illustration of these operations is given in Fig. 3.9.

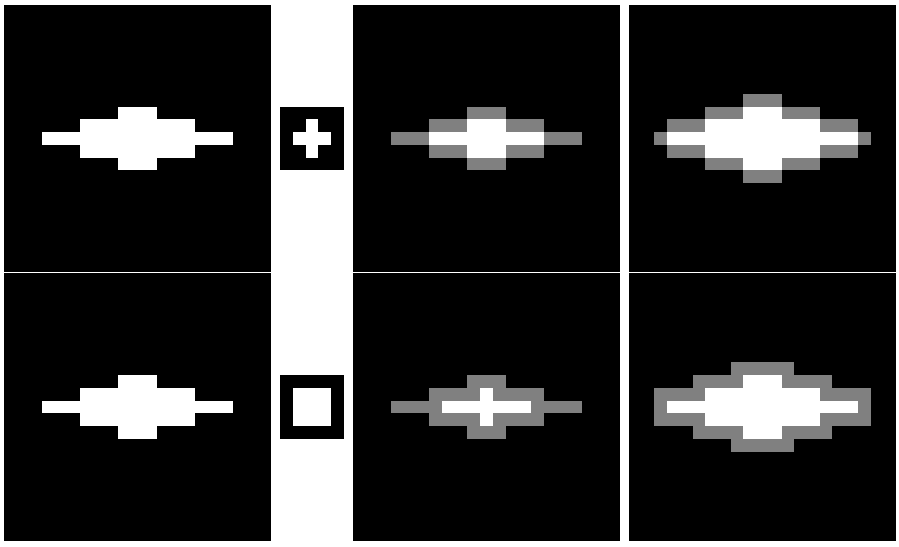


Figure 3.9: Examples of erosion and dilation, the most common mathematical morphology operations. In both row, from left to right: input binary image, structuring element, erosion of the input image by the structuring element, dilation of the image by the structuring element. Gray pixels denote pixels set to 0 in the erosions and pixels set to 1 in the dilations.

Common morphological operations also include opening and closing, which are combinations of erosion and dilation:

$$\text{Opening: } \gamma_S(I) = \delta_S(\epsilon_S(I)) \tag{3.12}$$

$$\text{Closing: } \phi_S(I) = \epsilon_S(\delta_S(I)). \tag{3.13}$$

These operations can be extended to grayscale images. If I is the image function, assigning a real value to each pixel p, then the operations are defined as:

$$\text{Erosion: } \epsilon_S(I) = \inf_{q \in D_I} [I(q) - S(q-p)] \tag{3.14}$$

$$\text{Dilation: } \delta_S(I) = \sup_{q \in D_I} [I(q) + S(p-q)], \tag{3.15}$$

where D_I is the domain of I and S is the function defining the structuring element. Usually, flat structuring elements are used:

$$S(p) = \begin{cases} 0 & \text{if} p \in E \\ -\infty & \text{otherwise} \end{cases} \tag{3.16}$$

where $E \subseteq D_I$.

3.2.2 Application to source detection

Aptoula et al. (2006) use the morphological smoothing OCCO filter (Peters, 1995) with a disk structuring element as a preprocessing step. OCCO stands for Open-Close Close-Open and is defined as:

$$\text{OCCO}_S(I) = \left\lfloor \frac{1}{2}\gamma_S(\phi_S(I)) + \frac{1}{2}\phi_S(\gamma_S(I)) \right\rfloor. \tag{3.17}$$

Then, a watershed transform (Beucher and Lantuejoul, 1979; Beucher and Meyer, 1993) is applied. The name refers to geology topography, where different high reliefs separate drainage basins. In image processing, the watershed transform handles grayscale images like topographic maps and finds ridge lines using virtual "flooding" to achieve image segmentation. It is also used by Zheng et al. (2015) to divide the image in sub-regions around bright stars to tune the rest of the analysis on each particular sub-region.

In a simpler way, Yang et al. (2008) use an opening with a circle structuring element to extract the sky background component after a value-stretching operation to evenly distribute image pixel values. Classical detection techniques (see section 3.1) are then applied to extract sources.

After some classical preprocessing, Perret et al. (2009) use the hit-or-miss transform, a widely used pattern recognition operation in mathematical morphology. Given two structuring elements H and M with $H \cap M = \emptyset$, the hit-or-miss transform of I is:

$$HM(I) = (I \ominus H) \cap (I_C \ominus M), \tag{3.18}$$

where I_C is the complement of the set I. The result is given by the points that fit in H and do not fit in M, hence the name.

Berger et al. (2007); Baillard et al. (2007) propose an algorithm that computes the component tree of an image. This is a representation of an image where the child relations between the tree nodes define spatial inclusions while nodes at the same level in the tree represent connected components. This representation makes possible to identify objects and the authors present a quick application to astronomical images. Connected trees are also used in Perret et al. (2010).

Yet, all those methods have limitations similar to the ones described in 3.1. They are all based on heuristics, need to manage parameters and are tuned for a particular cases. Furthermore, they are almost all combined with the basic detection methods described in Section 3.1.

3.3 Multiscale approaches

Another interesting category of source detection techniques is the multiscale approach. One major drawback of the basic detection techniques described above is that they do not handle very well objects that show up at different scales in images, especially galaxies. Multiscale methods try to overcome this by detecting objects at different scales.

The general principle of multiscale approaches is to decompose the images into components at different scales and to detect objects that stand out the most from the noise at those scales. Most multiscale methods are based on wavelet decompositions. In the following, I will focus on such methods because these are the most popular but other multiscale techniques exist, such as the pyramidal median transform (Starck et al., 1999), or iterative Gaussian smoothing (Kaiser et al., 1995).

3.3.1 Wavelet transform

Image decomposition is performed in a so-called wavelet space, where the wavelets are scaled and shifted versions of an analyzing wavelet ψ, or mother wavelet, which has zero mean:

$$\psi_{s,l}(x) = \frac{1}{s^{1/2}}\psi\Big(\frac{x-l}{s}\Big), \tag{3.19}$$

where s defines the scaling, l defines the shifting, $\psi_{s,l}$ are the wavelets, and ψ is the mother wavelet. Dyadic scales are generally used, i.e., two consecutive scales are related by a factor two. The wavelet coefficients representing a function f in the wavelet space are obtained by correlating the function with the wavelets:

$$c_{f,\psi}(s,l) = \langle f, \psi_{s,l}\rangle = \int_{\mathbb{R}} f(x)\psi_{s,l}(x)dx \tag{3.20}$$

One of the most used analyzing wavelet, especially in astronomy, is the Mexican hat. In two dimensions, it is defined as:

$$\psi(x,y) = \left(1 - \frac{1}{x^2+y^2}\right)e^{-\frac{1}{2}(x^2+y^2)} \tag{3.21}$$

Representations of the function and its corresponding wavelets in one dimension are shown in Fig. 3.10.

In practice with images, the shifting parameter is just equivalent to moving each wavelet at each pixel. A different scaling can be used along each of the two dimensions, but it is rarely done because most objects to detect are isotropic. Thus, wavelet analysis often consist of correlating the image with different scaled wavelets. It results in different maps which correspond to filtered versions of the original image by the scaled wavelets.

3.3.2 Applications in astronomy

Wavelets have been extensively used in astronomy, especially for detecting galaxies, where several scales help to detect the different structures, and with X-ray and high energy imaging. For instance, Slezak et al. (1990) apply it in galaxy clusters and Damiani et al. (1997); Freeman et al. (2002) with X-ray images, all using a Mexican hat mother wavelet. Very recently, wavelets are still used for source detection in X-ray images (Nanni et al., 2020). Other methods have used b-spline interpolations (Unser and Aldroubi, 1992) as mother wavelet, like the multiscale vision model of Bijaoui and Rué (1995) or Lazzati et al. (1999); Slezak et al. (1994); Peracaula

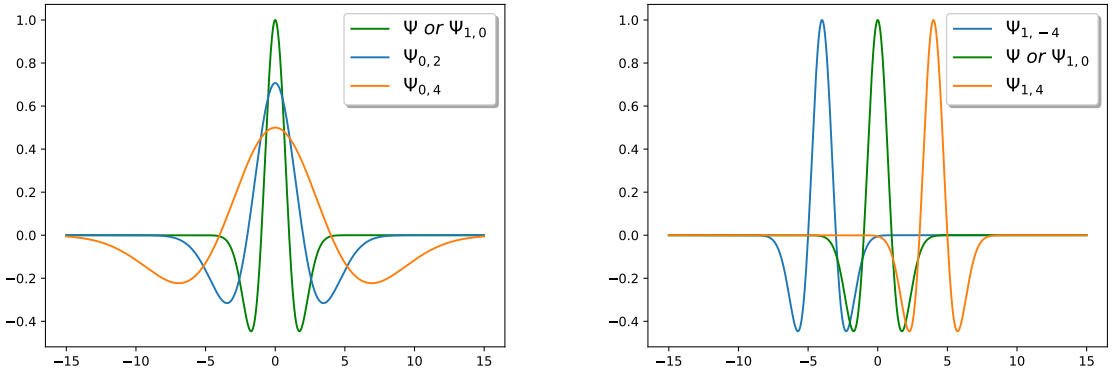


Figure 3.10: Some Mexican hat wavelets in one dimension. Green curves are the same and represent the mother wavelet in each graph. In the left graph, it is represented with two scaled versions of it. In the right graph, there are two shifted versions of the mother wavelet.

et al. (2011), the latter searching for extended structures in images. An example of wavelet decomposition of an image is shown in Fig. 3.11.

A significance level can be computed in each scale. See for example Starck and Pierre (1998). Afterwards, methods similar than those seen in Section 3.1 for thresholding and segmentation can be applied in each scale. A simple approach consists of detecting each source in its maximal appearance scale. More complex reconstructions are needed when one wants to recover the full object structures, as in Bijaoui and Rué (1995). However, there has never been a proper reconstruction scheme to fuse the extracted multiscale source components, which prevented multiscale approaches to be used in practice. Among the applications that were used for production one can mention: the identification of point-like sources in the cosmic microwave background (Cayón et al., 2000) in the context of the Planck mission and the detection of faint sources in ISOCAM data (Starck et al., 2003).

Multiscale approaches based on wavelets are also limited when it comes to detecting anisotropic features such as lines, curves and edges in images. This issue has motivated research on other sets of functions such as ridgelets and curvelets, which are extensions of wavelets. Ridgelets include rotation as an additional transformation of the mother ridgelet. While they are better ways of representing lines, ridgelets still struggle with curves and curved edges. Curvelets have been designed to use ridgelets locally, at a scale small enough to approximate curves as straight lines. See Fadili and Starck (2009) for more details.

3.3.3 Sparse representations and compressed sensing

Compressed sensing is a framework where signals can be sparsely represented, with fewer samples than the Shannon and Nyquist sampling theory states, using sets of functions or dictionaries (Bobin et al., 2008). It is based on the compressibility property of the data, i.e., the existence of a dictionary where the signal is sparsely encoded (Starck and Bobin, 2009), and has mainly applied to data reconstruction or denoising. Sparse coding techniques were partly motivated by experiments such as Olshausen and Field (1996) who found that searching for a sparse coding representation of natural scene images would lead to filters similar to what is observed in the V1 cortex receptive fields. Some astronomical applications have been developed, like denoising with dictionary learning (Beckouche et al., 2013). However, when it comes to building a vision model that could be used, e.g., for detecting sources, these approaches suffer from the same limitations

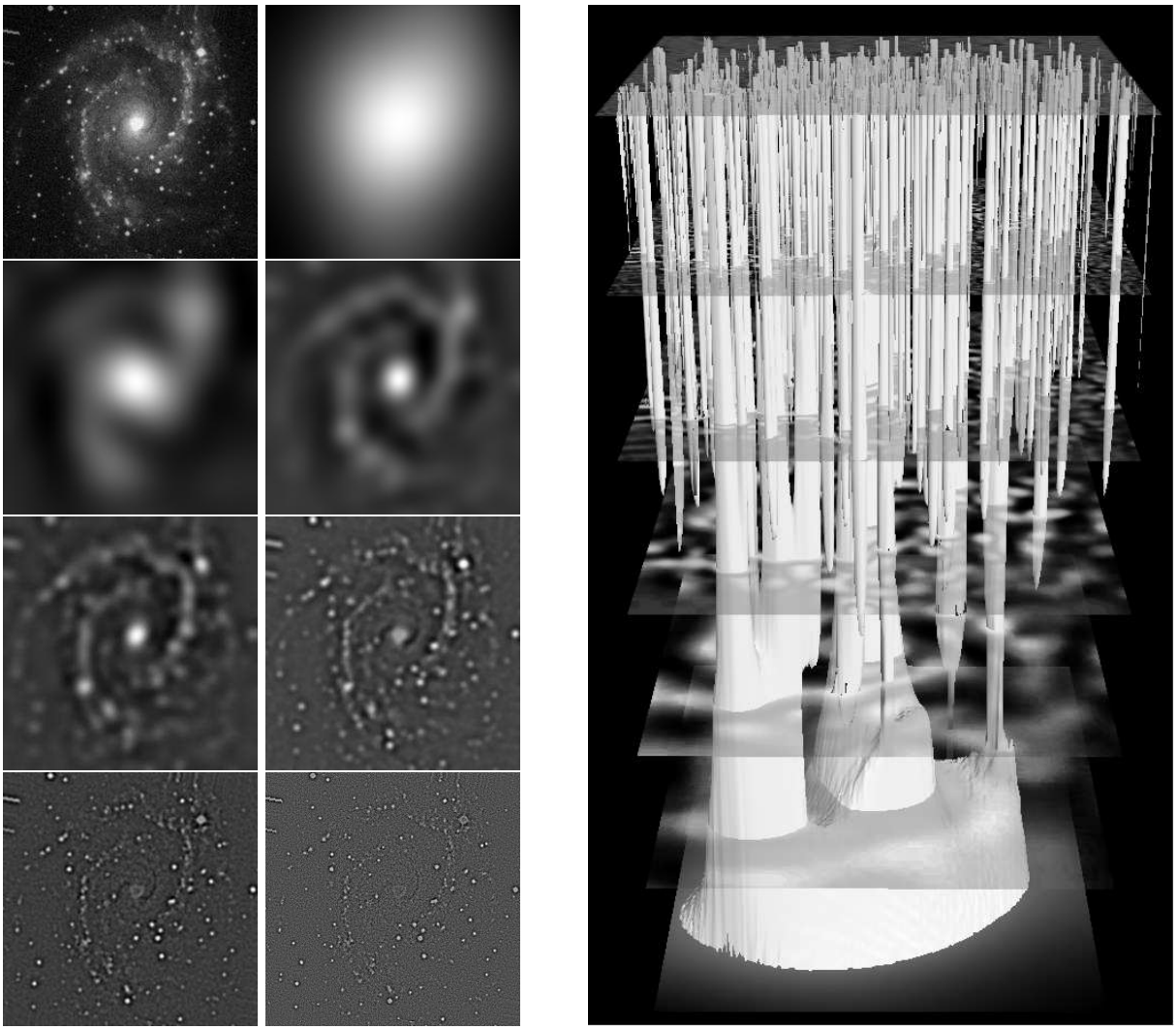


Figure 3.11: Left: a wavelet decomposition of a galaxy in seven scales. The top left image is the original galaxy image. Right: a tree built from the segmentation at each scale. Large scale features like the galaxy bulge appear at the bottom while small scale features like stars appear on the top. Images credits: www.multiresolution.com and Bertin (2001).

as multiscale methods: no generic solutions were found for reconnecting the extracted signal components.

3.3.4 Probabilistic catalogs

Before concluding, we must mention a particular point of view which is that of probabilistic catalogs. Hobson and McLachlan (2003) present two ways to detect sources. Firstly, an iterative method that stops using a Bayesian evidence criterion. This method has later been optimized (Carvalho et al., 2009, 2012). Secondly, a method detecting all objects at once, tested on a toy problem, linking with probabilistic catalogs.

More recently, Bayesian statistics have been used for source detection via probabilistic cataloging. It consists of inferring catalogs as posteriors and was first designed by Brewer et al. (2013). It has been applied to optical (Portillo et al., 2017), X-ray (Jones et al., 2015) and gamma (Daylan et al., 2017) data. An extension of Portillo et al. (2017) to multi-band data has been proposed in Feder et al. (2020).

3.4 Conclusion

The source detection methods currently in production mostly rely on (fixed) matched filtering and thresholding. Although this approach is the linear optimal solution for detecting isolated sources with known profiles (in the presence of stationary noise), it is not optimal in various other regimes where images are contaminated, crowded, or containing sources of various shapes and sizes. In order to adapt the detection to the latter issue, multiscale methods have been developed. They are mainly based on classical bases of functions such as wavelets and have evolved to sparse representations and the search for relevant data features.

By aspiring to use deep learning techniques, we follow a similar direction. Indeed, the essence of deep learning is to find abstract representations of the data to solve a particular task. Yet, this is a big paradigm change compared to classically designed algorithms because the representations are not handcrafted but directly learnt from raw data. The approaches that we will use now will consist of data-driven forward models instead of algorithms.

In the next chapter, I present the machine learning concepts that we will need for the rest of the manuscript.

Chapter 4

Feedforward neural networks applied to images: from the single neuron to convolutional neural networks

The purpose of this chapter is to introduce the terminology and concepts that we will use in the remaining of this manuscript. In the context of supervised machine learning, I review the main stages in the history of feedforward neural networks, from the most basic systems to the deep architectures used for modern image classification[1]. I focus on some aspects regarding activation functions, cost functions and regularization. Emphasis is put on the ability of neural networks to act as Bayesian classifiers under specific conditions. Finally, I introduce convolutional neural networks that are the supervised feedforward neural networks of choice when dealing with images.

4.1 Overview and supervised learning

In the scope of this work, feedforward neural networks are used as supervised learning systems. Supervised learning itself is already part of a bigger family which also includes unsupervised machine learning and reinforcement learning.

The gist of supervised learning with feedforward neural networks, represented in Fig. 4.1, is to fit a function that maps inputs to outputs based on a data set of known input-output pairs. This fitting process, known as learning or training, is an iterative process that uses a loss function (or cost function) and an optimization method: training steps are iterated over the data samples. At each learning step, predictions are made by the model, and a cost function evaluates how good or bad the predictions are. Based on this evaluation, the model parameters are updated using an optimization method to fit better the input-output pairs in the future. The optimization method is usually based on gradient descent (at first order) because it is the only tractable method for training large networks.

The two main tasks that are learnt by feedforward neural networks through supervised learning are classification, that is, deriving class membership, and regression, that is, fitting an arbitrary function between the input and the output.

The type of models that are used here are feedforward neural networks, i.e., neural networks where the connections between nodes are organized in layers and do not form a cycle. Within a layer, the node inputs are fed from the previous layer and their outputs are fed to the next layer, hence the name feedforward. There are no connections between the nodes of the same layer.

[1]For a thorough introduction to pattern recognition, machine learning and deep learning, see, e.g., Bishop (2006) and Goodfellow et al. (2016). For a more historical point of view, see, e.g., Schmidhuber (2015).

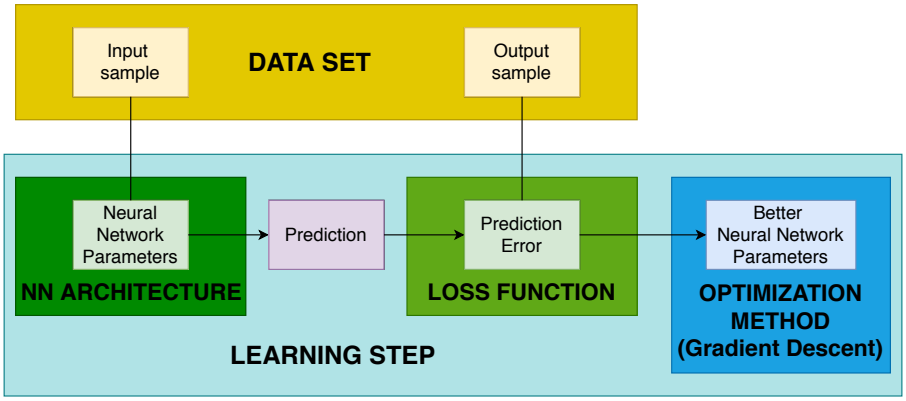


Figure 4.1: Scheme representing supervised learning with feedforward neural networks.

This differs from, e.g., Hopfield networks (Little, 1974; Hopfield, 1982) which are a form of recurrent neural networks where all nodes are interconnected, or Kohonen networks (Kohonen, 1982), also known as self organizing maps, which are unsupervised learning networks ruled by neighborhood constraints.

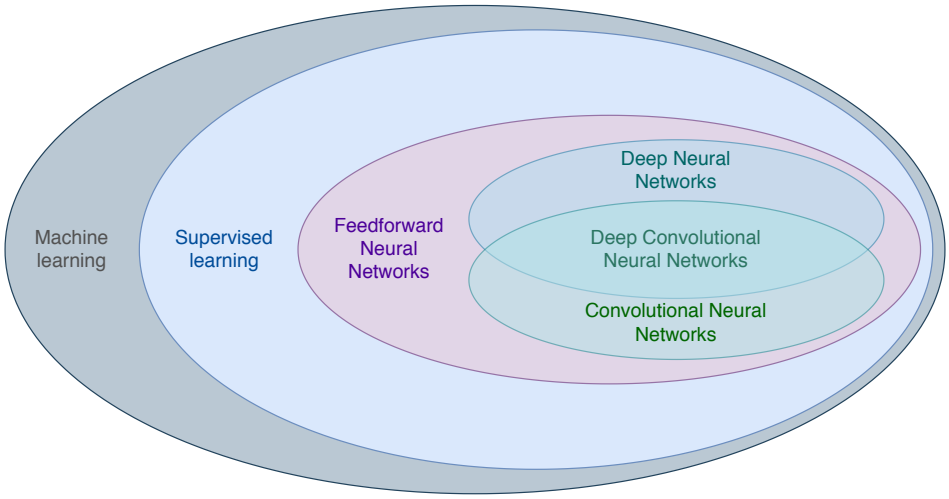


Figure 4.2: Diagram illustrating the place of feedforward neural networks in the machine learning systems. Note that some feedforward neural networks, such as autoencoders (Kingma and Welling, 2013), are unsupervised models.

Feedforward neural networks have then been extended to Convolutional Neural Networks (CNNs, LeCun et al., 1990), which are particularly suited to processing regularly sampled data such as images. CNNs have contributed to the rise of deep learning, using models with a large number of layers to capture more abstract features in the data. Fig. 4.2 gives a simplified overview of all these fields.

4.2 Feedforward neural networks and how they operate

Feedforward neural networks have been developed in several steps during the 20th century and the beginning of the 21st century.

4.2.1 The beginning of neural networks: the artificial neuron

Let us start by the beginning: the artificial neuron (McCulloch and Pitts, 1943) and the particular case of the Perceptron (Rosenblatt, 1958).

The neuron

A model of the artificial neuron is represented in Fig. 4.3. The behavior of the neuron is controlled by a weight vector $\boldsymbol{w}$. The neuron performs the dot product[2] of $\mathbf{w}$ with the input vector $\mathbf{x}$, adds a bias b, and passes the result through an activation function f.

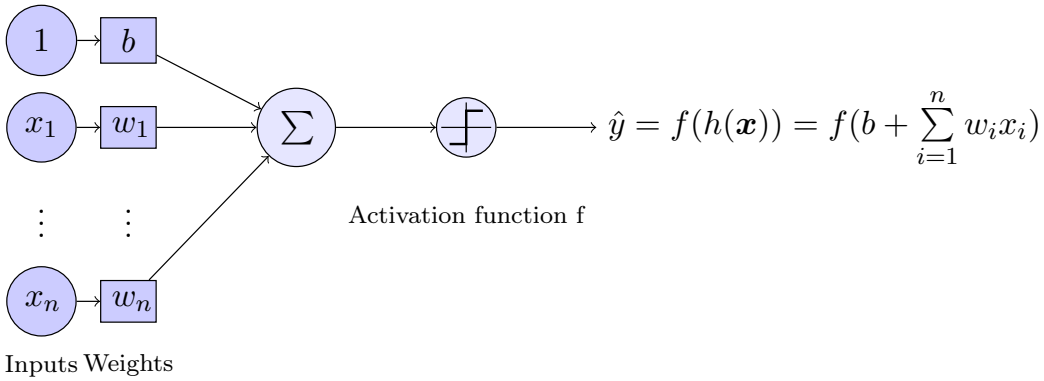


Figure 4.3: Schematic representation of the formal neuron.

Artificial neurons were originally conceived as simplified models of biological neurons. However, even though artificial neuron networks share characteristics with what is understood to happen in the brain, contemporary neural networks have become closer to pure machine learning models rather than brain models.

The Perceptron

The Perceptron model (Rosenblatt, 1958) is an artificial neuron with f being the Heaviside function, that is:

$$f(z) = \begin{cases} 0 & \text{if} \quad x < 0 \\ 1 & \text{if} \quad x \geq 0. \end{cases} \tag{4.1}$$

Or:

$$f(z) = \begin{cases} 0 & \text{if} \quad x < 0 \\ 0.5 & \text{if} \quad x = 0 \\ 1 & \text{if} \quad x > 0. \end{cases} \tag{4.2}$$

Although (Hebb, 1949) already proposed a neuron learning rule, Rosenblatt (1958) was probably the first to propose a training algorithm with a practical implementation. Given a set of N input-output pairs $(\boldsymbol{x}_k, y_k)$, Rosenblatt (1958) adjusts the weights w_i and the bias b iteratively by applying the following rules for each pair in the data set:

$$w_i^{(t+1)} = w_i^{(t)} + \eta(y_k - \hat{y_k})x_{k_i} \tag{4.3}$$

$$b^{(t+1)} = b^{(t)} + \eta(y_k - \hat{y_k}), \tag{4.4}$$

[2]Note that other combinations of weights and input are possible, e.g., radial basis functions (Broomhead and Lowe, 1988b,a).

where η, the learning rate, controls the amplitude of the weight updates. η is one of the model hyperparameters, i.e., variables that control the training process without being part of the trained parameters themselves (the weights and the bias). The number of training steps is another example of a hyperparameter. Fig. 4.4 shows an example of a training data set of 100 points $\boldsymbol{x}$ where $\boldsymbol{x} = (x_1, x_2)$ with two classes: points above and below $x_2 = x_1$.

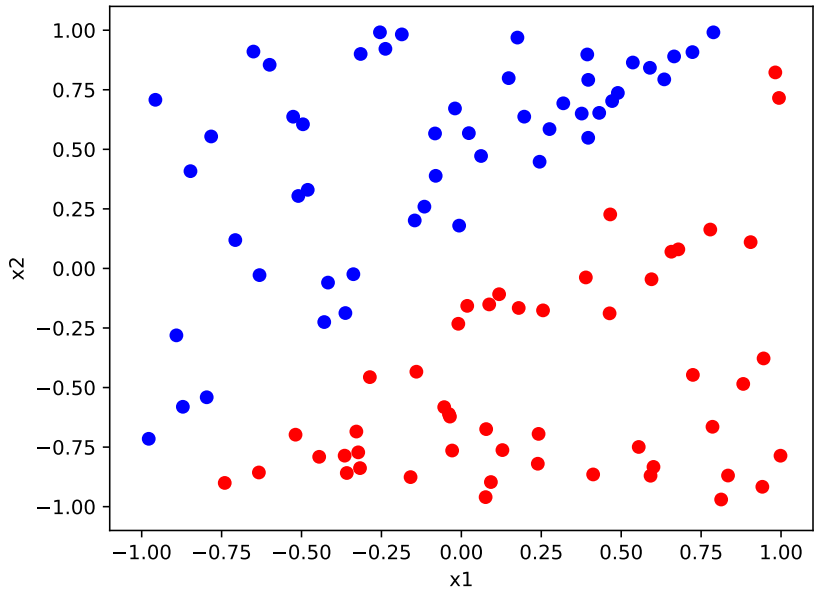


Figure 4.4: Example of training set with two classes.

Using this set to train a Perceptron with three parameters, w_1, w_2, b, one can reach a solution like the one shown in Fig. 4.5. In this case, convergence as defined in (Rosenblatt, 1958) is reached in 52 iterations using $\eta = 0.01$. Note that this can vary depending on η, the picking order of the data samples during training and the initial parameter values (0 for all parameters, following Rosenblatt (1958)). It is nevertheless common to apply random values at start, especially in multilayered neural networks (as we will see later), so that all neurons do not compute the same outputs.

Fig. 4.6 shows the evolution of the parameters and the accuracy during training. Weights are converging toward -0.018, 0.021, and 0.000 for $w_1, w_2, and b$, respectively, which makes sense as the Perceptron looks at the sign of $x_2 - x_1$ to separate the two classes.

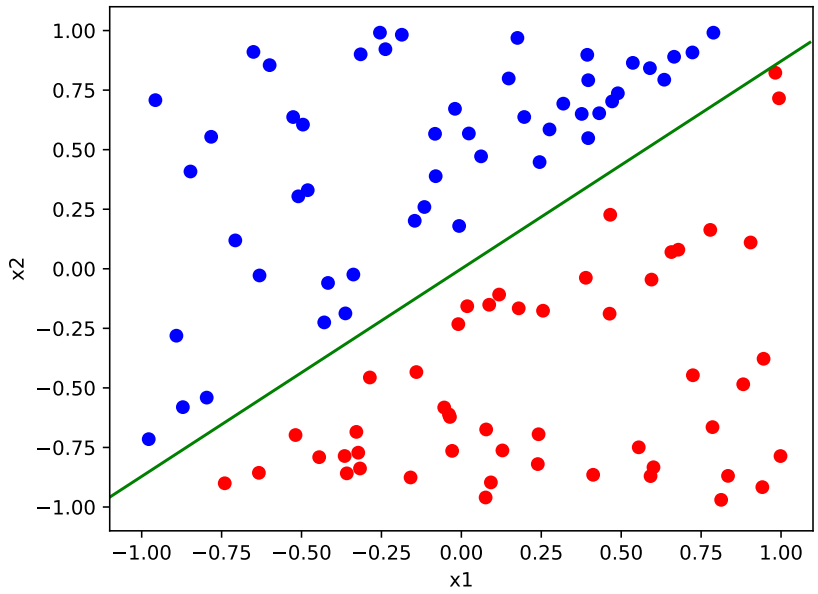


Figure 4.5: Linear separation learnt by a Perceptron.

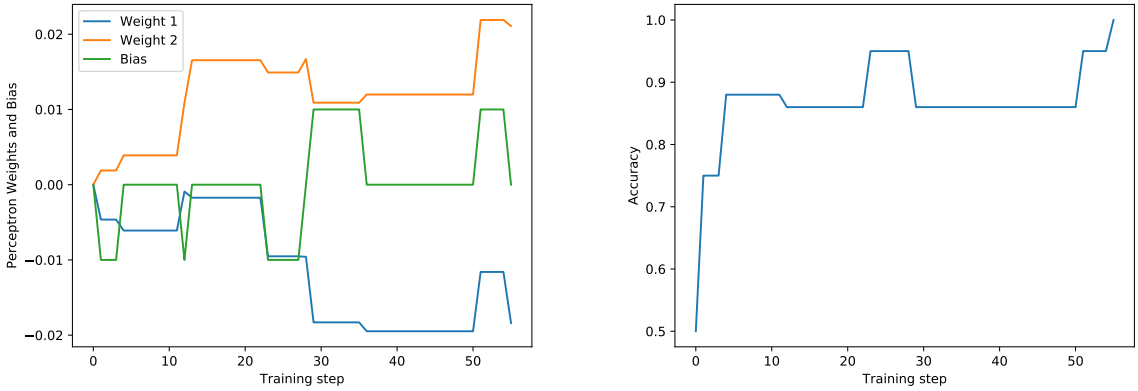


Figure 4.6: Evolution of Perceptron parameters and accuracy during training.

Learning with gradient descent

The continuous Perceptron (Rosenblatt, 1958) is a Perceptron that uses an activation function $f : \mathbb{R}^n \to [0, 1]$. It can be trained using a cost function and gradient descent. Gradient descent involves minimizing a function by iteratively adjusting the weights in the direction opposite to the error gradient. Given a function $f : \mathbb{R} \to \mathbb{R}$, an initial point x_0 is randomly picked, and at each step the following update:

$$x_{n+1} = x_n - \eta \frac{df}{dx}(x_n) \tag{4.5}$$

makes the function closer to its (local) minimum. Obviously, f must be differentiable. This is the basic gradient descent algorithm but a lot of variations and improvements exist, especially regarding convergence. In practice one uses more sophisticated gradient descent techniques. Ruder (2016) provides a good overview of the existing gradient descent algorithms.

For example, given the single variable function f so that $\forall x \in \mathbb{R}, f(x) = x^2$, the aim is to find x^* so that $f(x^*) = 0$. The process consists then of starting with a random x_0 and to use the rule $x_{n+1} = x_n - 2\eta x_n$. Fig. 4.7 illustrates this process using $\eta = 0.25$:

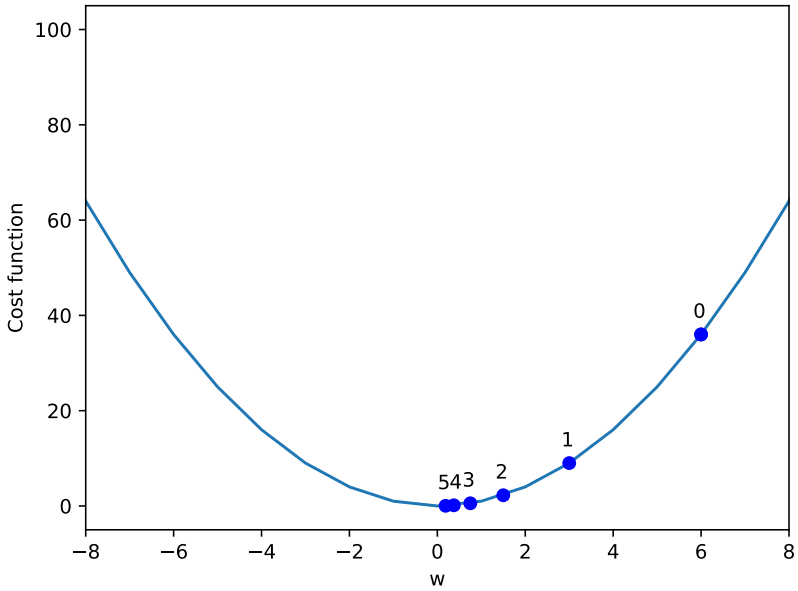


Figure 4.7: Example of a gradient descent. The algorithm starts from the point 0 and iterates to get closer to the minimum of the function.

Even in this simple example, a bad choice for η can prevent the method from converging satisfactorily: very high η values will make the cost oscillating between the two sides of the curve, while very low values will make the convergence excessively slow. Learning rate decay is a common way to address these issues.

Let's apply gradient descent to an artificial neuron. An artificial neuron makes a prediction for a single data sample $(\boldsymbol{x}_k, y_k)$ as follows:

$$\hat{y}_k = f(h(\boldsymbol{x}_k)) = f(b + \boldsymbol{w}.\boldsymbol{x}_k). \tag{4.6}$$

An example of a cost function is the squared error:

$$E_k(\boldsymbol{x}_k, \boldsymbol{w}, b) = \frac{1}{2}(\hat{y_k} - y_k)^2. \tag{4.7}$$

The gradient descent update writes:

$$w_i^{'} = w_i - \eta \frac{\partial E_k(\boldsymbol{x}, \boldsymbol{w}, b)}{\partial w_i} \tag{4.8}$$

$$b^{'} = b - \eta \frac{\partial E_k(\boldsymbol{x}, \boldsymbol{w}, b)}{\partial b}. \tag{4.9}$$

For writing convenience, the variable which E_k depends on can be omitted. Remember that E_k is a function of the input $\boldsymbol{x}$ and the parameters of the model. The gradient here is:

$$\frac{\partial E_k}{\partial w_i} = (\hat{y}_k - y_k) f'(h(\boldsymbol{x})) x_{i_k} \tag{4.10}$$

$$\frac{\partial E_k}{\partial b} = (\hat{y}_k - y_k) f'(h(\boldsymbol{x})). \tag{4.11}$$

This can be derived directly from Eq. 4.6 or using the chain rule:

$$\frac{\partial E_k}{\partial w_i} = \frac{\partial E_k}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial h} \frac{\partial h}{\partial w_i}, \tag{4.12}$$

with

$$\frac{\partial E_k}{\partial \hat{y}} = (\hat{y}_k - y_k) \tag{4.13}$$

$$\frac{\partial \hat{y}}{\partial h} = f'(h(\boldsymbol{x})) \tag{4.14}$$

$$\frac{\partial h}{\partial w_i} = x_{i,k}. \tag{4.15}$$

Using the chain rule may appear excessive here but we do so in anticipation of the optimization of multilayered neural network that we will see later.

So the update rules are:

$$w_i^{'} = w_i - \eta(\hat{y_k} - y_k) f'(h(\boldsymbol{x})) x_{i,k} \tag{4.16}$$

$$b^{'} = b - \eta(\hat{y_k} - y_k) f'(h(\boldsymbol{x})). \tag{4.17}$$

Such a process, iterating and updating the parameters sample by sample, is called online learning.

However the cost function can also be summed over the N input-output pairs of the data set:

$$E(\boldsymbol{x}, \boldsymbol{w}, b) = \sum_k^N E_k(\boldsymbol{x}, \boldsymbol{w}, b) = \sum_k^N \frac{1}{2}(y_k - \hat{y_k})^2, \tag{4.18}$$

and the gradient can be computed as:

$$\frac{\partial E}{\partial w_i} = \sum_k^N (y_k - \hat{y}_k) f'(h(\boldsymbol{x})) x_{i,k} \tag{4.19}$$

$$\frac{\partial E}{\partial b} = \sum_k^N (y_k - \hat{y}_k) f'(h(\boldsymbol{x})), \tag{4.20}$$

so that the update can be made for the all the data at once:

$$w_i^{'} = w_i + \eta \sum_k^N (\hat{y_k} - y_k) f'(h(\boldsymbol{x})) x_{i,k} \tag{4.21}$$

$$b^{'} = b + \eta \sum_k^N (\hat{y_k} - y_k) f'(h(\boldsymbol{x})). \tag{4.22}$$

This way of processing is called batch (or deterministic) learning. In practice, batch learning cannot be performed with very large data sets and can easily get stuck in local minima. On the other hand, online learning can lead to a very "noisy" learning process because of updates moving from one direction to another, especially if there are outliers (this may however be mitigated by using a very small learning rate).

Fortunately a compromise can be achieved through mini-batch learning[3], which consists in updating the weights using small subsets of the data. In the case of online and mini-batch

[3]Confusingly, mini-batch learning is often simply referred to as "batch learning". It is also common to refer to the size of the subsets used at each training step as the "batch size".

learning, gradient descent is called Stochastic Gradient Descent (SGD), because samples are randomly selected at each training step, unlike batch learning.

Batch gradient descent always converges to the minimum of a convex cost function in single-layer networks. Using stochastic approximation theory (Robbins and Monro, 1951; Blum et al., 1954), it can also be shown that convergence to local extrema is guaranteed for non-convex cost functions (Bottou, 1998)[4].

The Perceptron learning rule (Eq. 4.4) can be derived from Eq. (4.17). As the activation function f is monotonously increasing, the first derivative is positive and can be omitted from Eq. 4.17. Gradient descent provides a generic way to train an artificial neuron or a Perceptron. The only requirement is that the activation function must be differentiable[5].

Limitations of the Perceptron

By construction, a (first-order) Perceptron can only draw linear separations of the input space. In other words, it can only classify patterns that can be separated with a hyperplane in the input space (Minsky and Papert, 1969). For example, it fails to emulate the logical XOR operation, which is a non-linearly separable problem, as shown in Fig. 4.8.

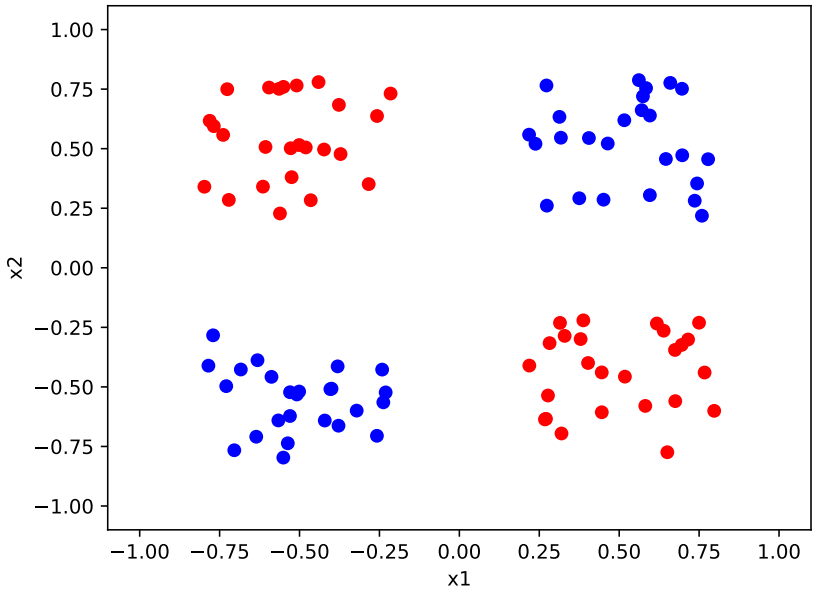


Figure 4.8: Example of an XOR-like problem.

Higher order Perceptrons have been proposed (e.g., Minsky and Papert, 1969), which in addition to the linear combination of inputs also combine products of inputs of degree 2, 3 or more:

$$\hat{y} = f\left(\sum_{i}^{N} w_i x_i + \sum_{i,\ j=1,\ i\leq j}^{N} w_{i,j} x_i x_j + \sum_{i,\ j,\ k=1,\ i\leq j\leq k} w_{i,j,k} x_i x_j x_k + \ \ldots\right). \tag{4.23}$$

[4]Latest version at https://leon.bottou.org/publications/pdf/online-1998.pdf

[5]Several popular activation functions such as ReLU do not meet this requirement in 0, but it does not impact stochastic gradient descent in practice.

This increases the capabilities of the Perceptron but it is limited to polynomial combinations of the inputs that are chosen beforehand.

Instead, more complex functions can be learnt by adding "hidden" neuron layers to obtain multilayered feedforward neural networks. The idea was already present in Minsky and Papert (1969), but doubts about the existence of a converging algorithm to train such systems were too high at the time.

4.2.2 Multilayered feedforward neural networks

The principle of multilayered feedforward neural networks is simply to insert one or several layers in the original Perceptron, which is why they are often referred to as MultiLayer Perceptrons (MLPs). In such networks, the inputs of a given layer are the outputs of the previous layer (except for the first layer, directly connected to the inputs).

Universal approximation theorem

Adding extra layers makes it possible to solve non-linearly separable problems, but how complex can these problems be ? The universal approximation theorem (Hornik et al., 1989; Cybenko, 1989) provides elements of answer. It states that, under some conditions, feedforward neural networks with at least one hidden layer are universal approximators, meaning that they can approximate any continuous function over a compact subset of the input space. The conditions are a linear output, a non-polynomial activation function in the hidden layer (such as the sigmoid or tanh, see below), and an arbitrary large number of neurons per layer. It was later shown that the defining constraint is not a specific category of activation functions, but rather the architecture of the network (Hornik, 1991).

Despite the convenient theoretical fact that feedforward neural networks with hidden layers *can* act as universal approximators, one must keep in mind that the number of neurons that can be managed in practice may be too small, and the training algorithm not powerful enough for some functions to be mapped with sufficient accuracy.

Forward pass in a multilayered feedforward neural network

Let's consider the multilayered feed forward neural network in Fig. 4.9:

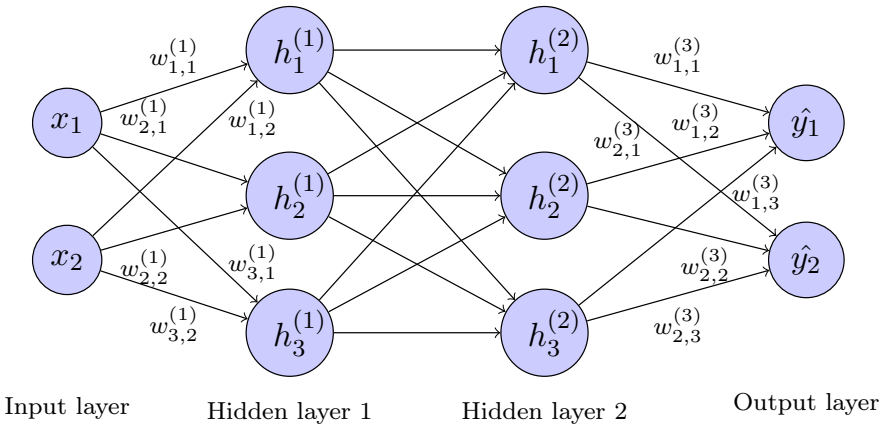


Figure 4.9: Diagram of a multilayered feedforward neural network.

This multilayered feedforward neural network takes a two-dimensional vector as input and returns another two-dimensional vector as output. It has 2 hidden layers with 3 artificial neu-

rons each. For clarity, the biases of all neurons and the weights connecting the two hidden layers are not represented in Fig. 4.9. These weights would be noted $w^{(2)}_{j,i}$, with i indicating the corresponding neuron in the first hidden layer while j would indicate the one in the second layer.

The following feedforward equations can be written to make a prediction:

$$\boldsymbol{o}^{(1)} = f_{h^{(1)}}(\mathbf{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}), \tag{4.24}$$

$$\boldsymbol{o}^{(2)} = f_{h^{(2)}}(\mathbf{W}^{(2)}\boldsymbol{o}^{(1)} + \boldsymbol{b}^{(2)}), \tag{4.25}$$

$$\hat{\boldsymbol{y}} = f_y(\mathbf{W}^{(y)}\boldsymbol{o}^{(\mathbf{2})} + \boldsymbol{b}^{(y)}), \tag{4.26}$$

where $f_{h^{(1)}}$, $f_{h^{(2)}}$, and f_y are the activation functions of the first hidden layer, second hidden layer, and output layer, respectively, and

$$\mathbf{W}^{(1)} = \begin{pmatrix} w^{(1)}_{1,1} & w^{(1)}_{1,2} \\ w^{(1)}_{2,1} & w^{(1)}_{2,2} \\ w^{(1)}_{3,1} & w^{(1)}_{3,2} \end{pmatrix}, \mathbf{W}^{(2)} = \begin{pmatrix} w^{(2)}_{1,1} & w^{(2)}_{1,2} & w^{(2)}_{1,3} \\ w^{(2)}_{2,1} & w^{(2)}_{2,2} & w^{(2)}_{2,3} \\ w^{(2)}_{3,1} & w^{(2)}_{3,2} & w^{(3)}_{3,3} \end{pmatrix}, \mathbf{W}^{(y)} = \begin{pmatrix} w^{(y)}_{1,1} & w^{(y)}_{1,2} & w^{(y)}_{1,3} \\ w^{(y)}_{2,1} & w^{(y)}_{2,2} & w^{(y)}_{2,3} \end{pmatrix} \tag{4.27}$$

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \boldsymbol{b}^{(1)} = \begin{pmatrix} b^{(1)}_1 \\ b^{(1)}_2 \\ b^{(1)}_3 \end{pmatrix}, \boldsymbol{b}^{(2)} = \begin{pmatrix} b^{(2)}_1 \\ b^{(2)}_2 \\ b^{(2)}_3 \end{pmatrix}, \boldsymbol{b}^{(y)} = \begin{pmatrix} b^{(y)}_1 \\ b^{(y)}_2 \end{pmatrix} \tag{4.28}$$

$$\boldsymbol{o}^{(1)} = \begin{pmatrix} o^{(1)}_1 \\ o^{(1)}_2 \\ o^{(1)}_3 \end{pmatrix}, \boldsymbol{o}^{(2)} = \begin{pmatrix} o^{(2)}_1 \\ o^{(2)}_2 \\ o^{(2)}_3 \end{pmatrix}, \hat{\boldsymbol{y}} = \begin{pmatrix} \hat{y_1} \\ \hat{y_2} \end{pmatrix}. \tag{4.29}$$

Backpropagation for multilayered neural network learning

Keeping the squared error cost function, we have:

$$E(\boldsymbol{x}, \theta) = \frac{1}{2}\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|^2_2, \tag{4.30}$$

where $\boldsymbol{x}$ is the input sample, $\boldsymbol{y}$ its "ground truth" output, $\hat{\boldsymbol{y}}$ is the prediction and $\| \ \|_2$ denotes the l2 norm, that is:

$$\|\boldsymbol{z}\|_2 = \sqrt{\sum_i^n z_i^2}. \tag{4.31}$$

From what was seen in 4.2.1, it is quite straightforward to update the parameters $\mathbf{W}^{(3)}$ and $\boldsymbol{b}^{(3)}$. The gradient descent method states:

$$w'^{y}_{j,i} = w^{(y)}_{j,i} - \eta\frac{\partial E}{\partial w^{(y)}_{j,i}} \tag{4.32}$$

$$b'^{y}_{j} = b^{(y)}_{j} - \eta\frac{\partial E}{\partial b^{(y)}_{j}}, \tag{4.33}$$

where:

$$\frac{\partial E}{\partial w^{(y)}_{j,i}} = \frac{\partial E}{\partial \hat{y_j}}\frac{\partial \hat{y_j}}{\partial h^{(y)}_j}\frac{\partial h^{(y)}_j}{\partial w^{(y)}_{j,i}}, \tag{4.34}$$

with:

$$\frac{\partial E}{\partial \hat{y_j}} = (y_j - \hat{y_j}) \tag{4.35}$$

$$\frac{\partial \hat{y_j}}{\partial h_j^{(y)}} = f'_y(h_j^{(y)}(\boldsymbol{x})) \tag{4.36}$$

$$\frac{\partial h_j^{(y)}}{\partial w_{j,i}^{(y)}} = o_i^{(2)}(\boldsymbol{x}). \tag{4.37}$$

Using the same process for the bias, the following update rules can be obtained:

$$w'^{y}_{j,i} = w_{j,i}^{(y)} - \eta(y_j - \hat{y_j}) f'_y(h_j^{(y)}(\boldsymbol{x})) o_i^{(2)}(\boldsymbol{x}) \tag{4.38}$$

$$b'^{y}_{j} = b_j^{(y)} - \eta(y_j - \hat{y_j}) f'_y(h_j^{(y)}(\boldsymbol{x})), \tag{4.39}$$

which are quite similar to those of the single neuron in Eq. 4.17. The only difference is that the update is guided by the output of the previous layer $o_i^{(2)}(\boldsymbol{x})$ instead of being directly connected to the input x_i.

Now, the weights of the hidden layers must be updated. Always with the gradient descent method:

$$w'^{(2)}_{j,i} = w_{j,i}^{(2)} - \eta \frac{\partial E}{\partial w_{j,i}^{(2)}} \tag{4.40}$$

$$b'^{(2)}_{j} = b_j^{(2)} - \eta \frac{\partial E}{\partial b_j^{(2)}}, \tag{4.41}$$

using the chain rule:

$$\frac{\partial E}{\partial w_{j,i}^{(2)}} = \frac{\partial E}{\partial o_j^{(2)}} \frac{\partial o_j^{(2)}}{\partial h_j^{(2)}} \frac{\partial h_j^{(2)}}{\partial w_{j,i}^{(2)}}, \tag{4.42}$$

with:

$$\frac{\partial E}{\partial o_j^{(2)}} = ??? \tag{4.43}$$

$$\frac{\partial o_j^{(2)}}{\partial h_j^{(2)}} = f'_{h^2}(h_j^{(2)}(\boldsymbol{x})) \tag{4.44}$$

$$\frac{\partial h_j^{(2)}}{\partial w_{i,j}^{(2)}} = o_i^{(1)}(\boldsymbol{x}). \tag{4.45}$$

In this case of an hidden layer, the first term is less straightforward than in the output layer where $o_j^{(2)}$ is in fact $\hat{y_j}$.

This is where the final error has to be backpropagated through the neural network to the current hidden layer. This is done by applying the chain rule a second time:

$$\frac{\partial E}{\partial o_j^{(2)}} = \sum_k^{n^{(y)}} \frac{\partial E}{\partial \hat{y_k}} \frac{\partial \hat{y_k}}{\partial h_k^{(y)}} \frac{\partial h_k^{(y)}}{\partial o_j^{(2)}} \tag{4.46}$$

$$= \sum_k^{n^{(y)}} (y_k - \hat{y_k}) f'_{h^{(y)}}(h_k^{(y)}(\boldsymbol{x})) w_{k,j}^{(y)}. \tag{4.47}$$

This is the so-called error backpropagation algorithm proposed by Rumelhart et al. (1985, 1988) (efforts along those lines can also be found in LeCun (1985b) LeCun (1985a), and Le Cun (1986)).

The output error is backpropagated to the previous layers. Note that the term $\frac{\partial E}{\partial h_k^{(y)}} = (y_k - \hat{y_k}) f'_{h^{(y)}}(h_k^{(y)}(\boldsymbol{x}))$ has already been computed when updating the weights of the output layers. But instead of being used with the output $o_{j,i}^{(2)}$ to update the weights of the output layer, these are backpropagated to the previous layer using $w_{k,j}^{(y)}$. So the rules are finally:

$$w'^2_{j,i} = w^2_{j,i} - \eta f'_{h^2}(h_j^{(2)}(\boldsymbol{x})) \sum_k^{n^{(y)}} \left((y_k - \hat{y_k}) f'_{h^{(y)}}(h_k^{(y)}(\boldsymbol{x})) w_{k,j}^{(y)} \right) \; o_i^{(1)}(\boldsymbol{x}) \tag{4.48}$$

$$b'^2_j = b^2_j - \eta f'_{h^2}(h_j^{(2)}(\boldsymbol{x})) \sum_k^{n^{(y)}} \left((y_k - \hat{y_k}) f'_{h^{(y)}}(h_k^{(y)}(\boldsymbol{x})) w_{k,j}^{(y)} \right). \tag{4.49}$$

For the first hidden layer, the same process is applied. The gradient descent method states:

$$w'^1_{j,i} = w^1_{j,i} - \eta \frac{\partial E}{\partial w_{j,i}^{(1)}} \tag{4.50}$$

$$b'^1_j = b^1_j - \eta \frac{\partial E}{\partial b_j^{(1)}}, \tag{4.51}$$

and:

$$\frac{\partial E}{\partial w_{j,i}^{(1)}} = \frac{\partial E}{\partial o_j^{(1)}} \frac{\partial o_j^{(1)}}{\partial h_j^{(1)}} \frac{\partial h_j^{(1)}}{\partial w_{j,i}^{(1)}}, \tag{4.52}$$

with:

$$\frac{\partial E}{\partial o_j^{(1)}} = \sum_k^{n^{h^2}} \frac{\partial E}{\partial o_k^{(2)}} \frac{\partial o_k^{(2)}}{\partial h_k^{(2)}} \frac{\partial h_k^{(2)}}{\partial o_j^{(1)}} \tag{4.53}$$

$$= \sum_k^{n^{h^2}} \frac{\partial E}{\partial o_k^{(2)}} f'_{h^{(2)}}(h_k^{(2)}(\boldsymbol{x})) w_{k,j}^{(1)} \tag{4.54}$$

$$\frac{\partial o_j^{(1)}}{\partial h_j^{(1)}} = f'_{h^1}(h_j^{(1)}(\boldsymbol{x})) \tag{4.55}$$

$$\frac{\partial h_j^{(1)}}{\partial w_{j,i}^{(1)}} = x_i, \tag{4.56}$$

where we know each $\frac{\partial E}{\partial o_k^{(2)}}$ from the previous update as $\sum_k^{n^{(y)}} (y_k - \hat{y_k}) f'_{h^{(y)}}(h_k^{(y)}(\boldsymbol{x})) w_{k,j}^{(y)}$. So finally the update rules are:

$$w'^1_{j,i} = w^1_{j,i} - \eta f'_{h^1}(h_j^{(1)}(\boldsymbol{x})) \sum_k^{n^{h^2}} \left(\frac{\partial E}{\partial o_k^{(2)}} f'_{h^{(2)}}(h_k^{(2)}(\boldsymbol{x})) w_{k,j}^{(1)} \right) x_i \tag{4.57}$$

$$b'^1_j = b^1_j - \eta f'_{h^1}(h_j^{(1)}(\boldsymbol{x})) \sum_k^{n^{h^2}} \left(\frac{\partial E}{\partial o_k^{(2)}} f'_{h^{(2)}}(h_k^{(2)}(\boldsymbol{x})) w_{k,j}^{(1)} \right). \tag{4.58}$$

One common and more convenient way to summarize all the update rules consists of using the following notations:

$$\frac{\partial E}{\partial w_{j,i}^L} = \delta_j^L o_i^{L-1} = \left(\frac{\partial E}{\partial o_j^L}\frac{\partial o_j^L}{\partial h_j^L}\right)\left(\frac{\partial h_j^L}{\partial w_{j,i}^L}\right), \tag{4.59}$$

with:

$$\delta_j^L = \frac{\partial E}{\partial o_j^L}\frac{\partial o_j^L}{\partial h_j^L} = \begin{cases} (y_j - \hat{y_j}) f_L'(h_j^L) & \text{if L is the last layer} \\ f_L'(h_j^L) \sum_k^{n^{L+1}} \delta_k^{L+1} w_{k,j}^L & \text{if L is an hidden layer} \end{cases} \tag{4.60}$$

$$o_i^L = \frac{\partial h_j^{L+1}}{\partial w_{j,i}^{L+1}} = \begin{cases} x_i & \text{if L is the first layer} \\ o_i^L & \text{if L is not the first layer} \end{cases} \tag{4.61}$$

so that the update rules are simply written:

$$w_{j,i}^{'L} = w_{j,i}^L - \eta \delta_j^L o_i^{L-1} \tag{4.62}$$

$$b_j^{'L} = b_j^L - \eta \delta_j^L. \tag{4.63}$$

Then, a learning step consists of 3 passes over the whole network:

- Forwarding the input through the whole network.
- Backpropagating the error by computing all the δ_j^L terms.
- Updating all the parameters $w_{j,i}^L$ and b_j^L.

The performance of the backpropagation algorithm relies on three important ingredients: activation functions, cost functions and regularization. Let us now review each of them. As we will see, important developments have occurred over the past ten years that have significantly improved learning performance.

4.2.3 Activation functions

Various functions can be used as an activation functions in a feedforward neural network. However in practice some functions are preferred for several reasons.

Most common activation functions

Here is a list of the most commonly used activation functions:

$$\text{Identity function: } f(h) = h \tag{4.64}$$

$$\text{Logistic or Sigmoid: } f(h) = \sigma(h) = \frac{1}{1+e^{-h}} \tag{4.65}$$

$$\text{Hyperbolic tangent: } f(h) = tanh(h) = \frac{e^h - e^{-h}}{e^h + e^h} \tag{4.66}$$

$$\text{Rectified linear unit: } f(h) = \begin{cases} 0 & \text{if} \quad h < 0 \\ z & \text{if} \quad h \geq 0. \end{cases} \tag{4.67}$$

$$\text{Softmax: } f_i(\boldsymbol{h}) = \frac{e^{h_i}}{\sum_j e^{h_j}} \tag{4.68}$$

Choosing the right activation functions

Several factors must be taken into account when choosing the activation functions of the various layers.

The ReLU (Rectified Linear Unit) function is currently one of the most commonly used activation function in hidden layers. It has been shown to run faster and provide better performance for neural networks with many layers (Jarrett et al., 2009; Nair and Hinton, 2010; Glorot et al., 2011), compared to the sigmoid and tanh activation functions that were used in the past. The latter saturate in both directions and their derivatives get very close to 0 for large positive or negative values. Weights are updated proportionally to the partial derivative of the error function, hence this "vanishing gradient" results in very slow updates and neurons getting stuck for specific combinations of weights. Such neurons are called "dead" neurons. The problem gets worse as the number of layers increases. The ReLU activation function partially solves this issue as it does not saturate on the positive side. However a ReLU neuron may still "die" if it gets in a state where the output is 0 for any input. Variants to the ReLU have been proposed to mitigate this problem, such as the leaky ReLU (Maas et al., 2013), which produces small negative values for negative inputs, softplus (Dugas et al., 2001), and ELU (Clevert et al., 2015, Exponential Linear Unit,):

$$\text{Leaky ReLU: } f(h) = \begin{cases} ah & \text{if} \quad h < 0 \\ h & \text{if} \quad h \geq 0. \end{cases} \tag{4.69}$$

$$\text{Softplus: } f(h) = \ln(1 + \exp(h)), \tag{4.70}$$

$$\text{ELU: } f(h) = \begin{cases} b(\exp(h) - 1) & \text{if} \quad h < 0 \\ h & \text{if} \quad h \geq 0 \end{cases} \tag{4.71}$$

The a parameter of the leaky ReLU can be made a trainable parameter (parametric ReLU, or PReLU, He et al., 2015).

Maxout is another trainable activation function (Goodfellow et al., 2013):

$$\text{Maxout: } f_i(\boldsymbol{h}) = \max_{j \in [1,k]} h_{ij}. \tag{4.72}$$

For example, in a layer composed of n neurons fully connected to the inputs, the idle output $\boldsymbol{h}$ is of dimension n. This idle output $\boldsymbol{h}$ can be divided into $\frac{n}{k}$ groups of k idle outputs. The maxout activation function takes the maximum values within each group: the i indices refer to $\frac{n}{k}$ while the j indices refer to k. The outputs are linear within each of the $\frac{n}{k}$ groups of k idle outputs. Maxout can be trained to combine these outputs and form a piece-wise linear approximation of a (convex) activation function. One drawback it that extra parameters are required for making the groups.

The choice of the activation functions for the last layer depends on the task at hand. For regression the identity function (linear neurons) is often the best choice. For mutually exclusive multiclass problems, the softmax function (Bridle, 1990; Jacobs et al., 1991), when paired with the proper cost function (see below), provides positive outputs that sum to 1 (Eq. 4.68) and facilitates the convergence toward posterior probabilities.

4.2.4 Cost functions

The cost function, or loss function, is one of the most important ingredients in neural network training. The most common cost functions are the quadratic cost and the cross entropy; they are used for regression and classification problems, respectively.

Quadratic cost

The quadratic cost is defined as the sum of squared differences between predictions and ground truth outputs, as seen in Section 4.2.1:

$$C = \frac{1}{2N} \sum_{k}^{N} \|\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k\|_2^2, \tag{4.73}$$

where:

- N is the batch size at the given training step.
- $\boldsymbol{y}_k$ is the k-th ground-truth vector.
- $\hat{\boldsymbol{y}}_k$ is the k-th predicted vector.
- $\|\|_2$ denotes the l2 norm, as presented in Eq. 4.31.

From a statistical standpoint, minimizing the quadratic cost function is equivalent to maximizing the likelihood that the predicted output matches the output data, assuming that the latter are subject to Gaussian errors (e.g., Bishop et al., 1995). The main inconvenient of the quadratic cost function is that it is very sensitive to outliers and mislabeled samples.

More general cost functions, build on arbitrary norms, may be used. For instance, the Minkowski error

$$C = \frac{1}{2N} \sum_{k}^{N} |\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k|^R \tag{4.74}$$

is the natural cost function for generalized Gaussian distributions (Bishop et al., 1995). $R = 2$ corresponds to the quadratic cost. Using $R < 2$ reduces the sensitivity to outliers.

Cross entropy

Cross entropy (Hopfield, 1987; Baum and Wilczek, 1988; Solla et al., 1988; Hinton, 1989; Rubinstein, 1999) is the cost function of choice for classification problems. As an information theory measure it may be used to quantify the difference between two probability distributions, here the ground truth probabilities y_k and the predicted probabilities $\hat{y}_k$. For a binary classification problem, cross entropy is defined as:

$$C = -\frac{1}{N} \sum_{k}^{N} \Big(y_k \log(\hat{y}_k) + (1 - y_k) \log(1 - \hat{y}_k) \Big). \tag{4.75}$$

From a statistical standpoint, minimizing binary cross entropy is equivalent to maximizing the probability that the predicted output matches the output data in the two-class problem, assuming that samples are independent (Bishop et al., 1995). Contrary to quadratic cost, cross entropy does not have the same relative cost near small and large ground truth values. Therefore it is more successful at estimating small ground truth values, i.e., small probabilities. In addition, cross entropy combines well with sigmoid activations and makes the gradient easier to compute.

Softmax cross entropy is defined as:

$$C = -\frac{1}{N} \sum_{k}^{N} \sum_{c}^{C} y_{k,c} \log(\hat{y}_{k,c}), \tag{4.76}$$

where C is the number of classes. It is the "natural" cost function for mutually exclusive multiclass problems (just as binary cross entropy is for binary classification problems). It also pairs well with

softmax activations (Bishop et al., 1995), hence the name "softmax cross entropy" (or categorical cross entropy).

In non mutually exclusive multiclass problems, where a sample can belong to several classes at the same time, one can use sigmoid cross entropy, which is defined as:

$$C = -\frac{1}{N}\sum_{k}^{N}\sum_{c}^{C}\Big(y_{k,c}\log(\hat{y}_{k,c}) + (1-y_{k,c})\log(1-\hat{y}_{k,c})\Big). \tag{4.77}$$

It is the sum of the binary cross entropies of all outputs. Each output is treated independently as a binary classification task, using *sigmoid* activation functions. Hence the name: "sigmoid cross entropy".

4.2.5 Regularization

In this section, I introduce the concepts of overfitting and underfitting as well as common regularization techniques that tackle overfitting.

Generalization, model capacity, underfitting, overfitting and testing set

Regularization techniques aim at reducing a possible network overfitting. Overfitting happens when the network exhibits excellent performance on the training set but poor performance on data not used for training. In order to detect overfitting, a common practice is to divide the data set in two: a training set containing data samples used for training the neural network, and a testing set containing data samples not used for training. The fraction of the data used for testing typically ranges from 20 to 40 percent. After or during learning, one can compare the performance obtained on the training and testing sets to make sure that they roughly match, indicating that no significant overfitting occurs.

Overfitting occurs when the capacity of the network, that is its ability to recover complex relations between inputs and outputs, is higher than the complexity of the task it must solve. In the extreme case where the network has more free parameters than training samples, learning "by heart" may occur. Such a trained network is very likely to perform poorly on test data as it does not catch the underlying trends. It is illustrated in Fig. 4.10.

On the contrary, underfitting happens if the capacity of the network is too low for the task complexity. The network is unable to fully solve the task.

From the statistical standpoint, the compromise between underfitting and overfitting (perfect training) represents a trade-off between bias and variance for the estimator formed by the trained network (see,e.g., Goodfellow et al., 2016).

These concepts are strongly related to the notion of generalization, which is the holy grail of machine learning. Generalization is the ability of the network to perform well not just on training data, but also on new, unseen data.

Overfitting can be particularly problematic when dealing with complex tasks that necessitate large networks. In such cases, it becomes mandatory to use regularization techniques that can help mitigating overfitting.

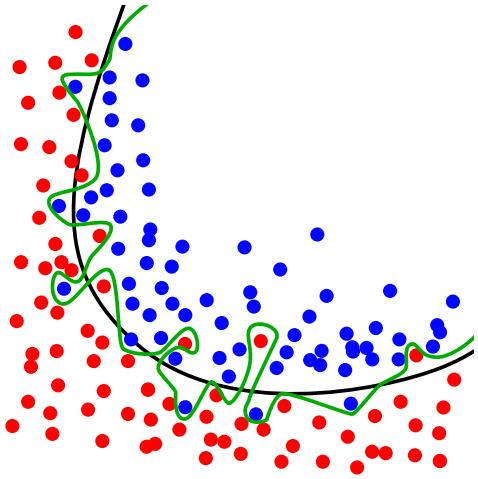


Figure 4.10: Example of overfitting. The underlying separation between blue and red points is drawn in black. An overfitting network learns the green delineation that perfectly classifies each training point but does not catch the underlying trend of the data. Image credits: *Wikipedia*.

l2-norm regularization

One of the simplest regularization techniques is l2-norm regularization, also known as Tikhonov regularization, ridge regression (Hoerl and Kennard, 1970) or weight decay (Plaut et al., 1986). l2-norm regularization was shown empirically to improve generalization in complex tasks (Hinton, 1987). It consists in adding to the cost function a contribution of the l2-norm of the network weight parameters, i.e., the following term:

$$L2_{reg} = \lambda \sum_{w \in \mathcal{W}} \|\boldsymbol{w}\|_2, \tag{4.78}$$

where $\mathcal{W}$ is the set of network weights and λ a hyperparameter defining the regularization strength. The value of λ may differ from one layer to the next (MacKay, 1992).

When using a quadratic cost, the effect of l2-norm regularization is to shrink the weights that do not contribute significantly to the cost function (see Goodfellow et al. (2016) for a demonstration). In a Bayesian framework, l2-norm regularization can be interpreted as a Gaussian prior on the weights (e.g., Bishop, 2006).

l1-norm regularization

Another way of penalizing weight values is l1-norm regularization (Tibshirani, 1996). Similarly to the l2 norm, it consists in adding the l1-norm of the network weight parameters to the cost function:

$$L1_{reg} = \lambda \sum_{w \in \mathcal{W}} \|\boldsymbol{w}\|_1, \tag{4.79}$$

where $\mathcal{W}$ is the set of network weights, λ is a hyperparameter that defines the strength of the regularization, and $\| \ \|_1$ denotes the l1-norm, defined as:

$$\|\boldsymbol{z}\|_1 = \sum_{i}^{n} |z_i| \tag{4.80}$$

When combined with a quadratic cost function, l1-norm regularization favors sparse solutions (Goodfellow et al., 2016). This sparsity property may also be used to enhance feature selection. l1 and l2 norm regularizations behave slightly differently, as shown in Fig. 4.11.

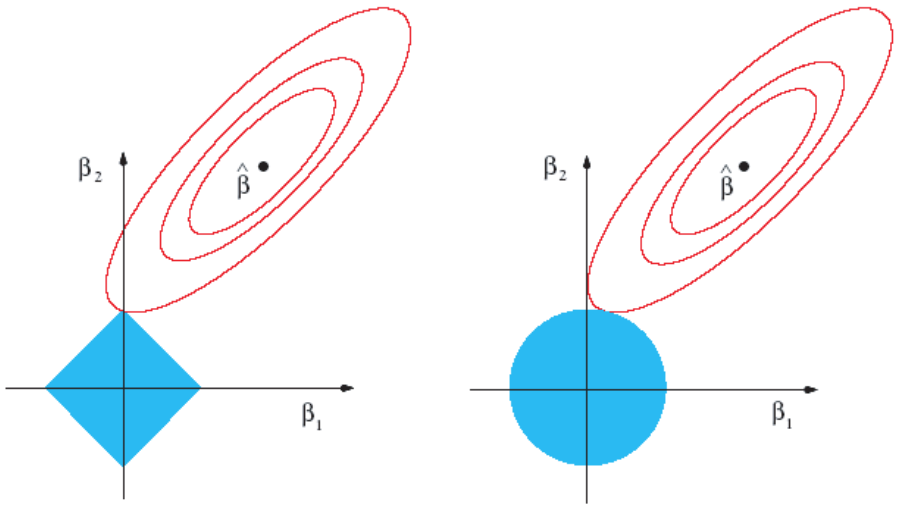


Figure 4.11: Illustration of l1 and l2 norm regularizations in a 2-dimensional weight space. Red ellipses trace identical cost values, and $\hat{\beta}$ is the position where the (unregularized) cost is minimal. Blue areas illustrate the constraint brought by l1 and l2 norm regularization: $|\beta_1| + |\beta_2| \leq R$ and $\beta_1^2 + \beta_2^2 \leq R^2$. Image credit: Hastie et al. (2009).

Early stopping

One may think that the longer the training, the better the performance, as the network keeps learning. In most cases, training and testing errors are both decreasing with the number of iterations. However the testing error may start increasing at some point, while the training error keeps decreasing: the network starts overfitting the training set. Early stopping is a common technique to avoid this. It simply consists in stopping the training before the network starts overfitting. First uses of this technique were found in Morgan and Bourlard (1990) and Weigend et al. (1990). An illustration is shown in Fig. 4.12.

To decide when to stop, a common practice is to use a validation data set. The validation set is a subset of the training set. It usually represents about 20 percent of the training set. It is used during training, but not for optimizing the model parameters. Instead, its purpose is to check for overfitting during training in order to tune the network hyperparameters (e.g., training duration). Training must generally be stopped when the validation loss starts increasing again. One may wonder whether the testing set could not be used for that. As a matter of fact, when possible, the testing data set should be used only for assessing the performance of the network, and not influence the training, including hyperparameter tuning. An additional criterion for early stopping is the training loss itself: it may be preferable to stop training if a plateau is reached.

Bishop et al. (1995) and Sjöberg et al. (1995) give clues as to why early stopping acts as a regularizer: stopping the training procedure prevents the network from using all its degrees of freedom and reaching a higher complexity. It forces some of the parameters to remain close to the initial values. One can show that early stopping acts as an l2-norm regularization on a quadratic cost function.

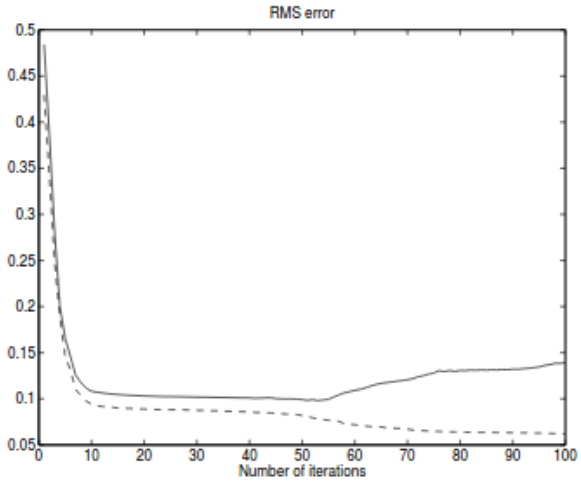


Figure 4.12: Illustration of overfitting due to overtraining. At some point the training error (solid line) is still decreasing while the validation error (dashed line) starts increasing. Image credit: Sjöberg et al. (1995).

In practice, even if early stopping is simple and efficient, it remains essentially empirical. The validation loss may be noisy, and there does not exist a specific rule to decide precisely when to stop Weigend et al. (1990); Sjöberg et al. (1995); Prechelt (1998).

Ensemble methods and dropout

The dropout regularization technique (Hinton et al., 2012; Srivastava et al., 2014) consists of ignoring a fraction of the units of a given layer by setting them to zero. The given fraction becomes a hyperparameter of the network and the ignored units set to zero are randomly picked at each training step. Doing so, only a subset of the layer is trained at each training step. The dropout algorithm shares similarities with ensemble methods (e.g., training a network on different subsets of the data and averaging the predictions at test time, a technique originally known as bagging Breiman (1996), or more generally model averaging). The idea behind ensemble methods is that several models may not fail on the same samples, i.e., that their errors may be uncorrelated. Averaging their result leads to better performance. However, this can be very expensive and not feasible with very large networks.

Instead, the purpose of the dropout method is to prevent neurons from co-adapting. By decoupling neurons that activate the same output together, the dropout technique makes each neuron more robust by forcing it to operate on its own. It has been empirically shown to greatly reduce overfitting and overcome other regularization techniques on various tasks (Srivastava et al., 2014). Interestingly, dropouts lead to a sparse solution, with fewer neurons being active at the same time (Srivastava et al., 2014).

The dropout technique has its drawbacks; it often requires more neurons and more training iterations. It does not work well with very small data sets, and is outperformed by other methods in this regime. Yet, it is computationally cheap and is now one of the most popular regularization techniques.

Data augmentation and noise injection

The regularization techniques we have described so far work by modifying the training procedure. Nevertheless, one may also act at the data level to reduce overfitting and improve generalization.

Some of the following techniques can be seen as data augmentation techniques, that is, methods that generate *new* training samples by modifying existing ones. For instance, in an image classification task, one may want to identify objects regardless of scale, location or orientation. This can be done by creating a new sample from a rescaled, shifted or rotated version of a pre-existing image, without changing the label. Such a process makes the network become invariant under the applied transformations.

Data augmentation is widely used for image classification as it is very easy to set up and achieves convincing results (Baird, 1995; Yaeger et al., 1997; Krizhevsky et al., 2012; Wang and Perez, 2017; Taylor and Nitschke, 2017). See Cao and Chen and Cabrera-Vives et al. (2017) for astronomical applications. Another data augmentation technique applicable to images is to erase random parts of images to increase robustness to occlusion (Zhong et al., 2017).

More sophisticated means to augment data involve dedicated neural networks to generate new samples: GANs (Generative Adversarial Networks, Goodfellow et al., 2014a; Brock et al., 2018) as used by, e.g., Antoniou et al. (2017) and Bowles et al. (2018), or style transfer networks (Gatys et al., 2015, 2016; Novak and Nikulin, 2016; Jing et al., 2019), as used by Mikołajczyk and Grochowski (2018) to create new data from existing data sets. The augmentation process may also be automated to optimize accuracy with the validation data set (Lemley et al., 2017; Tran et al., 2017; DeVries and Taylor, 2017; Cubuk et al., 2019).

Adding noise to the input data is also used as a regularization technique to make the model more robust to small changes in the input. Tiny, targeted changes to the input data have been shown to disrupt the results of image classification by a neural network (Szegedy et al., 2013). These new image samples are called adversarial examples. They cannot be distinguished from the original samples with the naked eye. Goodfellow et al. (2014b) hypothesize that neural networks have close to linear behavior for very small changes in input: this makes it possible to craft a small perturbation with a specific pattern, which thanks to a high dimensional weight vector can lead to large changes in output. Adding a targeted noise pattern to the inputs can be a way to make the network more robust to this type of attack (adversarial training). Increasing the generalization abilities is another motivation for adding a small amount of noise to the inputs (Sietsma and Dow, 1991). It has been shown to work as a regularization process in the case of a quadratic cost function (Bishop et al., 1995).

Noise can also be injected directly into the hidden neurons (Poole et al., 2014), or into network weights. It can be shown to act as a regularization process that makes the network parameters less sensitive to small perturbations (Goodfellow et al., 2016).

Finally, for classifiers one may also introduce changes to the ground truth used for training outputs. The process known as label smoothing (Szegedy et al., 2016) "softens" the initial ground truth values by replacing 0's and 1's with ϵ and $1 - \epsilon$, preventing the neural network to express overconfidence, especially in the presence of mislabeled samples. Interesting thoughts about label smoothing are presented in Müller et al. (2019).

Weight sharing techniques

The presence of a huge number of free parameters is often the cause of overfitting in neural networks, and reducing this number can have a strong regularization effect. Such a reduction may be achieved through weight sharing techniques. Weight sharing can happen in different ways. One possible way is to share some parts of the network to perform several tasks at the same time, a process known as multitask learning (Caruana, 1993). Such a type of sharing has been shown to improve generalization (Baxter, 1995). Another way to achieve this is to incorporate it directly in the network architecture, as we will see in Section 4.3.

Another kind of weight sharing is soft weight sharing (Nowlan and Hinton, 1992), which is a regularization technique where groups of weights are constrained to have similar values.

4.2.6 Estimating posterior probabilities

All the cost functions described in section 4.2.4 lead to an interesting property of neural network classifiers: in the conditions of perfect training, the outputs will estimate Bayesian posterior probabilities (Richard and Lippmann, 1991; Hampshire II and Pearlmutter, 1991; Miller et al., 1991; Rojas, 1996). This means that it should possible to adapt the outputs to new priors, i.e., new expected class proportions, after training. This is of great importance in many astronomical applications, where one has to deal with highly unbalanced data sets (e.g., when searching for rare events). Indeed, current neural network mini-batch training algorithms have issues when dealing with strong class imbalance (Japkowicz and Stephen, 2002; He and Garcia, 2009; Krawczyk, 2016; Khan et al., 2017). Minority classes can end up being "ignored", i.e., the classifier converges to a solution where it assigns all samples to the majority class. In order to avoid this behavior and reach a proper convergence, one must train the neural network with a more balanced data set. As a consequence, the training data do not reflect the expected class proportions (prior membership probabilities) found in the real data.

Fortunately, if one can assume that the trained classifier behaves as a perfect Bayesian classifier, i.e., that it returns the posterior probability $P(\omega_c|\boldsymbol{x}, T)$ of an input vector $\boldsymbol{x}$ to be a member of the class ω_c when trained on the training set T, one may update the output probabilities with the correct priors so that they better reflect the expected class proportions. From Bayes' rule $P(\omega_c|\boldsymbol{x}, T)$ can be written:

$$P(\omega_c|\boldsymbol{x}, T) = \frac{L(\boldsymbol{x}|\omega_c)P(\omega_c|T)}{\sum_i L(\boldsymbol{x}|\omega_i)P(\omega_i|T)}, \tag{4.81}$$

where $L(\boldsymbol{x}|\omega_c)$ is the likelihood of a sample $\boldsymbol{x}$ with class ω_c, and $P(\omega_c|T)$ is the prior probability of any training sample with class ω_c, that is, the fraction of samples with class ω_c in the training set. If all classes are equally represented, we have:

$$P(\omega_c|\boldsymbol{x}, T) = \frac{L(\boldsymbol{x}|\omega_c)}{\sum_i L(\boldsymbol{x}|\omega_i)}. \tag{4.82}$$

Now, with the real (observed) data we have:

$$P(\omega_c|\boldsymbol{x}, O) = \frac{L(\boldsymbol{x}|\omega_c)P(\omega_c|O)}{\sum_i L(\boldsymbol{x}|\omega_i)P(\omega_i|O)}, \tag{4.83}$$

where the $P(\omega_c|O)$'s may differ a lot from class to class. We can rewrite $P(\omega_c|\boldsymbol{x})$ as:

$$P(\omega_c|\boldsymbol{x}, O) = \frac{P(\omega_c|\boldsymbol{x}, T)L_T(\boldsymbol{x})P(\omega_c|O)}{\sum_i P(\omega_i|\boldsymbol{x}, T)L_T(\boldsymbol{x})P(\omega_i|O)} = \frac{P(\omega_c|\boldsymbol{x}, T)P(\omega_c|O)}{\sum_i P(\omega_i|\boldsymbol{x}, T)P(\omega_i|O)}, \tag{4.84}$$

where $L_T(\boldsymbol{x}) = \sum_{\omega_c} L(\boldsymbol{x}|\omega_c)$. In the case where all classes are not equally represented in the training set, that is when:

$$L(\boldsymbol{x}|\omega_c) = P(\omega_c|\boldsymbol{x}, T)\frac{\sum_i L(\boldsymbol{x}|\omega_i)P(\omega_i|T)}{P(\omega_c|T)}, \tag{4.85}$$

we have:

$$P(\omega_c|\boldsymbol{x}, O) = \frac{P(\omega_c|\boldsymbol{x}, T)P(\omega_c|O)}{P(\omega_c|T)\sum_i P(\omega_i|\boldsymbol{x}, T)\frac{P(\omega_i|O)}{P(\omega_i|T)}}. \tag{4.86}$$

In the binary classification problem, i.e., a problem with two classes ω_c and $\bar{\omega_c}$ (not ω_c), we have:

$$P(\omega_c|\boldsymbol{x}, O) = \frac{P(\omega_c|\boldsymbol{x}, T)P(\omega_c|O)}{P(\omega_c|T)(P(\omega_c|\boldsymbol{x}, T)\frac{P(\omega_c|O)}{P(\omega_c|T)} + P(\bar{\omega_c}|\boldsymbol{x}, T)\frac{P(\bar{\omega_c}|O)}{P(\bar{\omega_c}|T)})}, \tag{4.87}$$

which may also be written:

$$P(\omega_c|\boldsymbol{x}, O) = \frac{P(\omega_c|\boldsymbol{x}, T)}{P(\omega_c|\boldsymbol{x}, T) + \frac{P(\bar{\omega_c}|O)}{P(\omega_c|O)} \frac{P(\omega_c|T)}{P(\bar{\omega_c}|T)} P(\bar{\omega_c}|\boldsymbol{x}, T)}. \tag{4.88}$$

5 and 6 will give us an opportunity to check whether this Bayesian approach works or not in practice.

4.2.7 Multi-layered neural networks in practice

In this section, I describe the practical use and limitations of multilayered neural networks.

Training aspects

LeCun et al. (2012) provide many useful training tips that remain essential today. Two important points stand out:

- Input pre-processing: it is recommended that input values be small and around zero. If for instance all inputs are positive, all the weight updates will have the same sign, leading to inefficient learning. Plus, neural networks use a combination of small and precise thresholds on values. It therefore advised to reduce the dynamic range of input data, which is naturally high in astronomical images.
- Weight initialization: weights should be initialized randomly within a small range so that they remain in the linear regime of the activation functions, especially when using tanh or sigmoid. Values should be small enough so that activation functions do not saturate but not too small to avoid very shallow gradients and slow updates.

Other points of interest are discussed in LeCun et al. (2012), like choosing the right learning rate and the behavior of gradient descent. Ruder (2016) provides more insights into the optimization algorithms.

The batch normalization algorithm (Ioffe and Szegedy, 2015) appeared after LeCun et al. (2012) was published. As the name suggests, it consists of normalizing the data between each network layer. The normalization aims to overcome potential changes of the distribution of data across the network (a phenomenon known as internal covariate shift) so that each layer does not need to adapt to possible shifts in the distribution. It has been shown to make the training procedure faster and more stable, even though the underlying reasons of this success are still debated (Santurkar et al., 2018).

Applications and limitations of multilayered neural networks

Multilayered neural networks were used in various industry tasks throughout the 1990s (Wong et al., 1997), including computer vision problems such as image recognition (Khotanzad and Chung, 1998), object pose estimation (Khotanzad and Liou, 1996), handwritten or spoken digit and letter recognition (Burr, 1988), and image compression (Qiu et al., 1993). A well known example in astronomy, still in use today, is SExtractor's star/galaxy classifier (Bertin and Arnouts, 1996).

Still, these applications operate on very simple images, most of the time binary images, and not directly on pixels. Indeed, multilayered neural networks do not handle high dimensional inputs like images very well. Connecting each neuron to every pixel requires a very high number of weights. Such a high number of free parameters would inevitably lead to overfitting, or a network that fails to converge.

One would use pre-computed image features as network inputs instead. For example, Khotanzad and Chung (1998) use image moments (Teh and Chin, 1988) and Burr (1988); Qiu et al. (1993) use discretized versions of the images, recoded in small dimensional spaces. These features are handcrafted and problem dependent. SExtractor's star galaxy classifier uses isophotal areas of the source, the maximum pixel value and the seeing. See Xu et al. (1992) for a list of features used in early works involving multilayered neural networks in computer vision.

PCA (Principal Component Analysis, Wold et al., 1987; Abdi and Williams, 2010) was first introduced by Pearson (1901), and can be used to select decorrelated features. It is used in NExt Andreon et al. (2000), one of the first astronomical source detectors based on multilayered neural networks.

Handcrafted features can be extremely efficient but they still have limitations. They suffer from the curse of dimensionality and quickly become computationally expensive. In addition, they must be designed with great care. In practice, the neural network (or another type of classifier) is often mostly limited by the ability of these handcrafted features to characterize the data in a way which is relevant to the given task.

As we will see in the next section, convolutional layers provide an elegant solution to this problem for sampled input data such as images or time sequences, by giving the neural network the ability to learn directly from the pixels or measurements.

4.3 Convolutional neural networks

As their name suggests, Convolutional Neural Networks (CNNs) are based on the convolution operation (see section 2.2), which serves as a feature extractor. As we already saw in section 3.1.2, convolution with an appropriate kernel has the ability to enhance a specific pattern in the image. An example of a convolution kernel acting as a basic edge detector is shown in Fig. 4.13.

$$H = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \qquad H = \frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

Figure 4.13: Left: source image. Center: spatial convolution applied to an image using an edge detector kernel. Right: image convolution using a Gaussian blurring kernel (Image credits: *Wikipedia*).

A CNN is a feedforward network containing computational layers. It may also contain the usual fully connected layers described in section 4.2.2 (where each neuron is connected to all the features, hence the expression "fully connected").

In a convolutional layer one takes advantage of weight sharing by having a single neuron moved across all locations in the image. It turns out that this way to process is equivalent to a convolution, where the neuron weights are the convolution kernels. As the same set of weights is used across the whole image, it becomes possible to process high dimensional inputs like image rasters with a very small number of adjustable parameters. In comparison, a traditional layer

would use an intractable number of weights to connect each neuron to all the image pixels. Convolutional layers inherit the intrinsic translation equivariance property of convolution, which is convenient for object detection tasks, where features should be detected independently of their position in images.

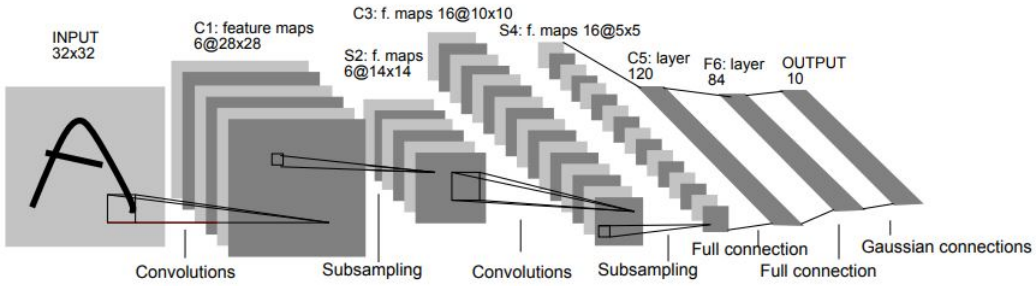


Figure 4.14: LeNet-5: a typical CNN architecture for image classification. Image credit: LeCun et al. (1998).

4.3.1 Basic architecture

The first convolution layer, computes multiple convolutions of the input data, called feature maps. A non-linear activation function is then applied element-wise to the convolved images. Once activated, a subsampling layer reduces the dimensions of the feature maps. Subsampling relies on pooling operations (Fig 4.15), which generally consist of taking the mean or the maximum of input tiles. Subsampling layers serve several purposes:

- Reducing the dimension of the feature maps, hence the number of parameters that are needed in the remaining part of the CNN. This reduces the computational cost and prevents potential overfitting.
- Allowing the convolutional kernels in subsequent layers to deal with larger spatial scales
- Introducing a level of translation invariance at small scales.
- Selecting the most important features at the current kernel scale.

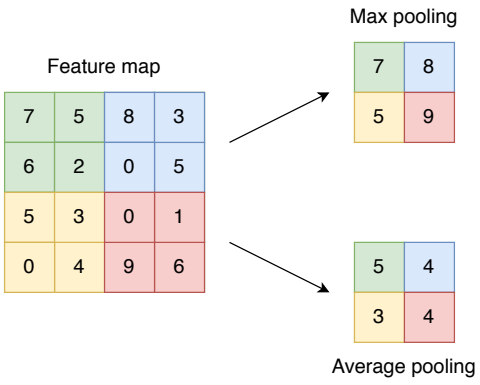


Figure 4.15: Examples of max pooling and average pooling applied to a 4×4 feature map. Since the 2×2 pooling kernel is moved every 2 pixels, pooling tiles are not overlapping and result in 2×2 feature maps. Note that the stride (step between two consecutive tiles) may not always be identical to the pooling kernel size, resulting in different output sizes.

CNNs generally stack several groups of convolutional, activation and subsampling layers[6]. Features are combined from one group to the next, allowing more and more abstract representations of the input data to be built. The succession of layers also makes the CNN able to process information over a wide range of spatial scales and capture more contextual information.

After several stacks of convolutional, activation and subsampling layers, features may be fetched into a series of fully connected layers to perform classification or regression tasks (Table 4.1).

Layer		Size	Kernel	Stride	Activation	#Parameters
Input	Image	$1 \times 32 \times 32$	-	-	-	-
C1	Conv	$6 \times 28 \times 28$	5×5	1	tanh	156
S1	Pool	$6 \times 14 \times 14$	2×2	2	-	-
C2	Conv	$16 \times 10 \times 10$	5×5	1	tanh	2416
S2	Pool	$16 \times 5 \times 5$	2×2	2	-	-
C5	Conv	$120 \times 1 \times 1$	5×5	1	tanh	48,120
F6	FC	84	-	-	tanh	10,164
Output	FC	10	-	-	softmax	850

Table 4.1: Table summing up the LeNet-5 (LeCun et al., 1998) CNN architecture for image classification. The output size of each layer is indicated as well as convolution and pooling kernel sizes, convolution and pooling stride sizes, activation functions and number of learnable parameters. Conv, Pool and FC stand for convolution layer, pooling layer (or subsampling layer) and fully connected layer, respectively.

4.3.2 Early CNN models

The first CNNs were designed for classifying handwritten digits (LeCun et al., 1989, 1990, 1995)[7]. The MNIST (Modified National Institute of Standards and Technology) database was used as a training set. MNIST is derived from an earlier NIST data set Grother (1995) and has been used for a long time as a benchmark, although it has since been overtaken by other more complex data sets. Fig. 4.14 shows LeNet-5, which is the archetype of basic CNN architectures for image classification.

Other early successful applications include face recognition Lawrence et al. (1997); Kwolek (2005); Osadchy et al. (2007) or speech recognition Sukittanon et al. (2004). The first application to natural scenes was done by Fu Jie Huang and LeCun (2006), who showed that CNNs could also learn features invariant under changing viewpoint and illumination. Their data set had 6 classes: human figures, four-legged animals, airplanes, trucks, cars and "none of the above".

In astronomy, an early convolutional model was developed in 1997 by E. Bertin: EyE [8], (Enhance your Extraction). EyE is a multilayered neural network connected to a moving window (retina). It was used for many years to identify cosmic-ray hits in large imaging surveys (e.g., Nonino et al., 1999).

While these are some early successes of CNNs, the latter have really started shining with the arrival of deep learning.

[6]The expression "convolutional layer" sometimes actually encompass the convolution, activation and subsampling layers.

[7]A convolutional architecture has already been proposed by Fukushima (1980), but it did not benefit from the efficient backpropagation algorithm.

[8]https://www.astromatic.net/software/eye

4.3.3 Deep learning models

With time, increasing computing power and the availability of larger labeled data sets allowed the scientists to start exploring more complex neural network models. In the early 2010's, deeper network models (i.e., models with more layers) became manageable and were found to allow much more abstract and complex representations of the data compared to previous, shallower models. For a comprehensive view of early deep learning techniques and machine learning, see, e.g., Goodfellow et al. (2016).

In the following, we will focus on the application of deep neural networks to image analysis, although deep nets have been applied to many other domains like audio classification Lee et al. (2009) or language processing Collobert and Weston (2008), just to name a few.

The ImageNet challenge

Even though other data challenges existed before, like the PASCAL VOC challenge (Everingham et al., 2010) that started in 2005, it is the ImageNet challenge, or ILSVRC (ImageNet Large Scale Visual Recognition Challenge), which provided the clear-cut demonstration of the superior performance of deep CNNs. The ILSVRC is an image classification challenge held since 2010 (Russakovsky et al., 2015). It is based on one of the largest image classification benchmark databases Deng et al. (2009). During the first two years, the two winning methods Lin et al. (2011); Sánchez and Perronnin (2011) were classical methods based on handcrafted feature extractors. The real turning point in the challenge was in 2012 with the arrival of AlexNet model (Krizhevsky et al., 2012), the first deep CNN to compete at the ILSVRC[9]. AlexNet won the challenge by a huge margin, suddenly decreasing the top-5 error rate from 25.8% in 2011 to 16.4%, while it was still 28.2% the year before (the top-5 error rate is measured by considering the five most probable classes predicted by the classifier).

The AlexNet architecture is shown in Fig. 4.16. It follows a classical CNN architecture as seen in Section 4.3.2. However it contains a larger number of convolution and pooling layers than LeNet-5. The last three fully connected layers are also much wider as there are $1,000$ classes. AlexNet contains more than 60 millions parameters, the majority being in the fully connected layers. Having as many parameters can lead to significant overfitting, which is mitigated with dropout Hinton et al. (2012) and data augmentation techniques (see section 4.2.5).

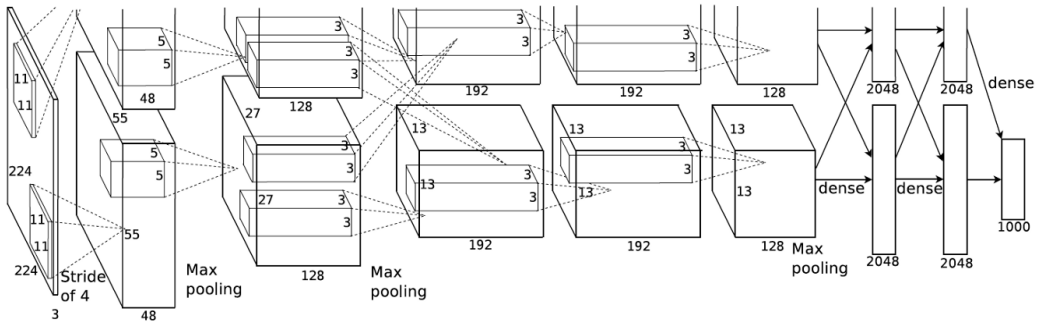


Figure 4.16: AlexNet was the first deep CNN to win the Imagenet challenge in 2012, decreasing the error rate from 25.8% to 16.4%. There are two data streams because computations were shared between two GPUs. Image credit: (Krizhevsky et al., 2012).

Since 2013, there has been an increase in the number of teams competing for ILSVRC, mostly

[9] A shallower CNN had already been run on ImageNet the year before (Ciresan et al., 2011).

using deep CNNs, and deep CNNs have since kept winning the challenge and decreasing the error rate, eventually surpassing humans.

Additional computer vision tasks have been held at the ImageNet challenge and won by deep CNNs (single object localization and object detection, where one must predict one bounding box per object class present in the image, or all the bounding boxes of all the objects present in the image, respectively). Over the years, the ILSVRC challenges have led to a lot of innovations and ideas for deep CNNs architectures. Some of the winning methods have become classical architectures used as bases to further works, sometimes referred as *backbone* networks. Two of this backbone networks are presented in the next section.

Classical deep CNN architectures for image classification

The first very deep CNN to win the Imagenet challenge was GoogLeNet (Szegedy et al., 2015) in 2014. GoogleNet is based on the Inception architecture, which consists of repeated building blocks called Inception modules (Appendix A.1). Each inception block concatenates feature maps obtained through different filterings of the input. The overall architecture of GoogLeNet is presented in Appendix A.2.

In 2014, GoogLeNet was in competition with VGG (Simonyan and Zisserman, 2014), which made an honorable second place in 2014 and has also become a widely used architecture. VGG is lighter than GoogleNet and uses smaller convolutional kernels. VGG-19 is described in Appendix A.2.

In 2015, the ILSVRC was won by ResNet (He et al., 2016). It uses residual blocks shown in Appendix A.1, where the feature maps of a given layer are reused again by addition to other feature maps downstream in the network. The overall architecture is shown in Appendix A.2. It was extended in Xie et al. (2017) by combining Inception-like blocks and skipped connections.

4.4 Conclusion

I have reviewed the history and the main concepts of feedforward and convolutional neural networks in the context of supervised learning and image classification. Thanks to the combination of clever training and regularization algorithms, deep convolutional neural networks have unquestionably become the most efficient algorithms for complex image recognition tasks. The superior efficiency of these models in computer vision tasks, combined with their ability to estimate posterior probabilities, matches well our requirements in terms of adaptability and robustness. This makes deep convolutional neural networks particularly attractive for our source detection project. In the next chapters, I discuss further state-of-the-art models going beyond image classification. I also come back to the Bayesian handling of neural network outputs when I present the practical applications that we have developed.

Chapter 5

Contaminant identification: MaxiMask and MaxiTrack

Astronomical images are far from perfect. A significant fraction of wide-field images of the deep sky are contaminated by defects (hereafter "contaminants"). These defects can easily trigger false detections, prevent detections or bias source measurements. The corresponding contamination introduced in the output source catalogs compromises the performance and scientific objectives of not only the COSMIC-DANCE and Euclid surveys, but also of many other surveys with strict science requirements. For example, in the Canada France Hawaii Telescope Lensing survey (Heymans et al., 2012), about 19% of the survey had to be discarded because of image defects.

Contaminants greatly complicate the source detection task, however none of the methods seen in Chapter 3 addresses them directly. Following our objective of robustness and universality, we thus aim at designing a tool capable of identifying contaminants prior to source detection. We want this tool to be capable of detecting a broad diversity of contaminants commonly found in astronomical images and to perform well under various optical and ambient conditions or detector properties, with minimal tuning.

In this chapter, I present our solution to this problem and its implemention in the form of two software packages: MaxiMask and MaxiTrack. I first review the various contaminants commonly affecting astronomical images that we chose to study. I describe how they are managed by existing methods. After describing the training data set, I present the CNN architectures of MaxiMask and MaxiTrack. I explain how we deal with strong class imbalance, which is a major challenge for MaxiMask, and how both packages can be operated in a Bayesian framework. I give a detailed report of the identification accuracy obtained on various types of test images and real data. Finally, I present the released packages[1] and conclude this chapter by discussing possible future developments.

The results of this study have been published in Astronomy & Astrophysics (Paillassa et al., 2020, , F).

5.1 Contaminants in astronomical images

Contaminants in astronomical images originate from various sources. We classify them in two categories: local and global contaminants. Local contaminants occur at the pixel level over a fraction of the image, while global contaminants affect the entire image.

All the contaminants presented hereafter are illustrated with images originating from the various instruments that have been used in our study. The references of these instruments are

[1] https://github.com/mpaillassa/maximask

given later when describing our work. The list of contaminants included here is not exhaustive. Nevertheless, it covers the most frequently found contaminants in images, including the COSMIC-DANCE and Euclid surveys.

5.1.1 Electronic contaminants

Electronic contaminants are local contaminants caused by defects or characteristics of the camera, which in our case can be a CCD or a CMOS, as described in Section 2.5.

Hot and dead pixels

Hot and dead pixels are the most common electronic contaminants. They come out as pixels having an anomalous response, either much higher or lower than expected, hence the names hot and dead, respectively. In most cases, they affect single pixels or columns because of the way CCDs are read. Yet, these can also appear as rows or as small clusters (Fig. 5.1).

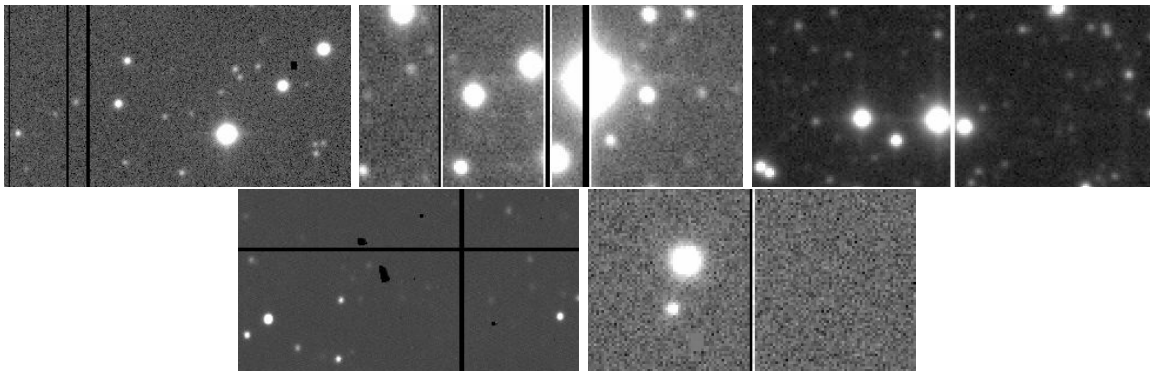


Figure 5.1: Examples of hot and dead pixel defects. Top: three examples from Megacam. Bottom left: an example from INT-WFC. Bottom right: an example from VST. Note that hot and dead columns are sometimes touching.

Saturated pixels and bleeding trails

Saturation and bleeding trails are local contaminants related to the detector's properties. Because the potential well in each pixel can only accumulate a limited number of electrons, the recorded value reaches a limit and stops increasing. In CCDs, wells can easily *overflow*, producing a saturation (or bleeding) trail along the direction of charge transfer. In CMOSes, the saturation pattern does not necessarily bleed and saturated pixels often exhibit non-physical values. Both types of saturation are shown in Fig. 5.2.

Figure 5.2: Left: typical bleeding trails in a Megacam image. Right: saturation pattern in a WFCAM image (infrared camera).

Persistence effect

Persistence is a local contaminant appearing as a remnant pattern caused by pixels exposed to a very bright star in a previous exposure that still emit signal. Persistence effects can vary a lot from a camera to another[2]. Examples are shown in Fig. 5.3.

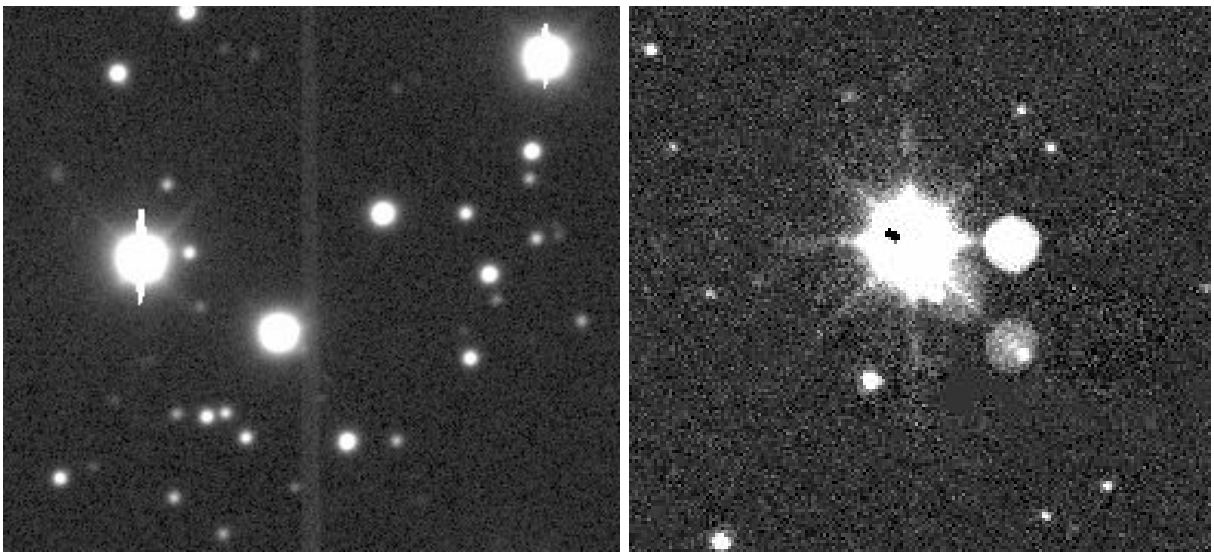


Figure 5.3: Left: example of persistence effects in a DECam image. Right: example of a persistence effect in a WFCAM image.

Crosstalk

Crosstalks are local contaminants related to the electronics. CCDs are sometimes divided in quadrants to improve readout speed, and each of them is in turn divided into channels corresponding to different reading ports. Since all the channel ports are read simultaneously, there

[2]See, e.g., http://casu.ast.cam.ac.uk/surveys-projects/wfcam/technical/persistence

can be some crosstalk, i.e., a channel reading output can be influenced by neighboring channels. Typically, a bright source in one channel will generate a donut-like pattern in the direction of the neighboring reading channels. An example is shown in Fig. 5.4. The pattern usually repeats at regularly spaced locations and becomes fainter at larger distances.

Figure 5.4: Example of a crosstalk pattern between channels in a WFCAM image.

Crosstalk can sometimes occur at the quadrant level. In this case, a bright star produces a negative mirror image with respect to the quadrant separation, as shown in Fig. 5.5.

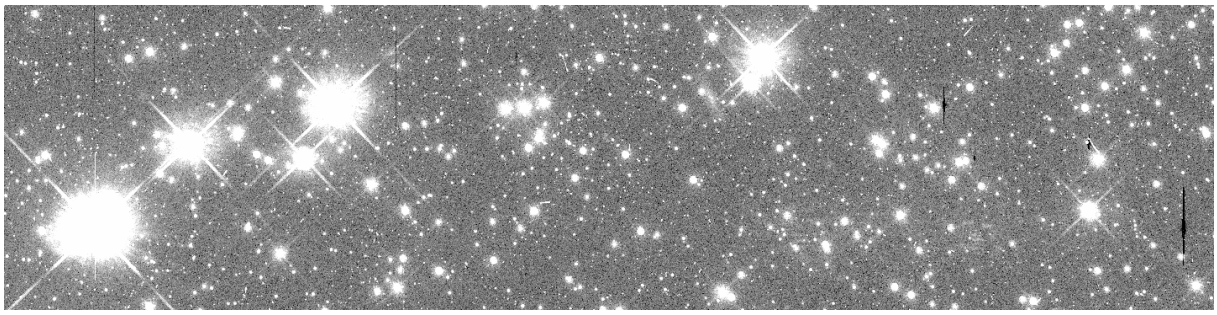


Figure 5.5: Example of crosstalk patterns between two quadrants in the WFC3 camera onboard HST. The separation between the two quadrants is in the middle of the image. Four bright stars from the left quadrant generate crosstalk patterns in the right quadrant. One star from the right quadrant produces a crosstalk pattern in the left quadrant.

5.1.2 Optical contaminants

Optical contaminants are caused by various features occurring in the optics of the telescope.

Residual fringing patterns

Fringes are local contaminants caused by thin-film interference in the detector. It is generated at the boundary between the electronics and the optics. The resulting patterns depend on the small detector thickness variations. Fringes are an additive feature and are generally removed right after the flat-fielding procedure thanks to fringing maps that are computed by combining a large number of exposures to be able to remove all the sources. Yet, the fringing map estimation and thus the subtraction procedure are not perfect and sometimes residual fringing patterns can still be found in images, as shown in Fig. 5.6.

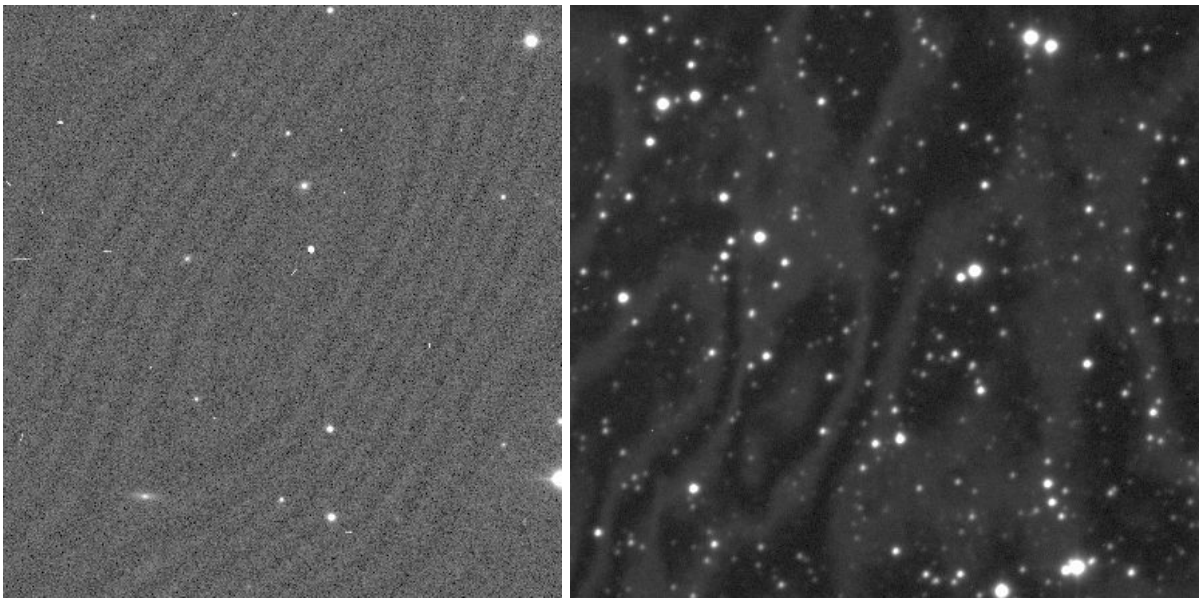


Figure 5.6: Left: fringing pattern residuals in a DECam image. Right: fringing pattern residuals in a Megacam image.

Diffraction spikes

Diffraction spikes are local contaminant appearing around bright stars caused by light diffraction from the spider supporting the secondary mirror. The shape is directly related to the geometry of the spider arms, and the number of spikes varies from one instrument to another. While many instruments exhibit four spikes in the form of a '+' or a 'x', others like WFCAM (Fig. 5.2) and Euclid (Fig. 5.8) have 3 spider arms, producing 6 diffraction spikes. This diversity adds another level of difficulty in our quest for a universal and generic tool capable of detecting spikes for any instrument. To add even more complexity, spikes can be variable in time or across the focal plane. In some instruments, diffraction spikes are indeed affected by a combination of various effects including distortions, telescope position, the presence of rough edges, cables around the spider arms, reflections on other telescope structures that make it variable. In the case of alt-azimuthal mounts the pattern will also rotate during an exposure. All of these variations can make the pattern change significantly and greatly complicate its identification. Typical examples of diffraction spikes are shown in Fig. 5.7.

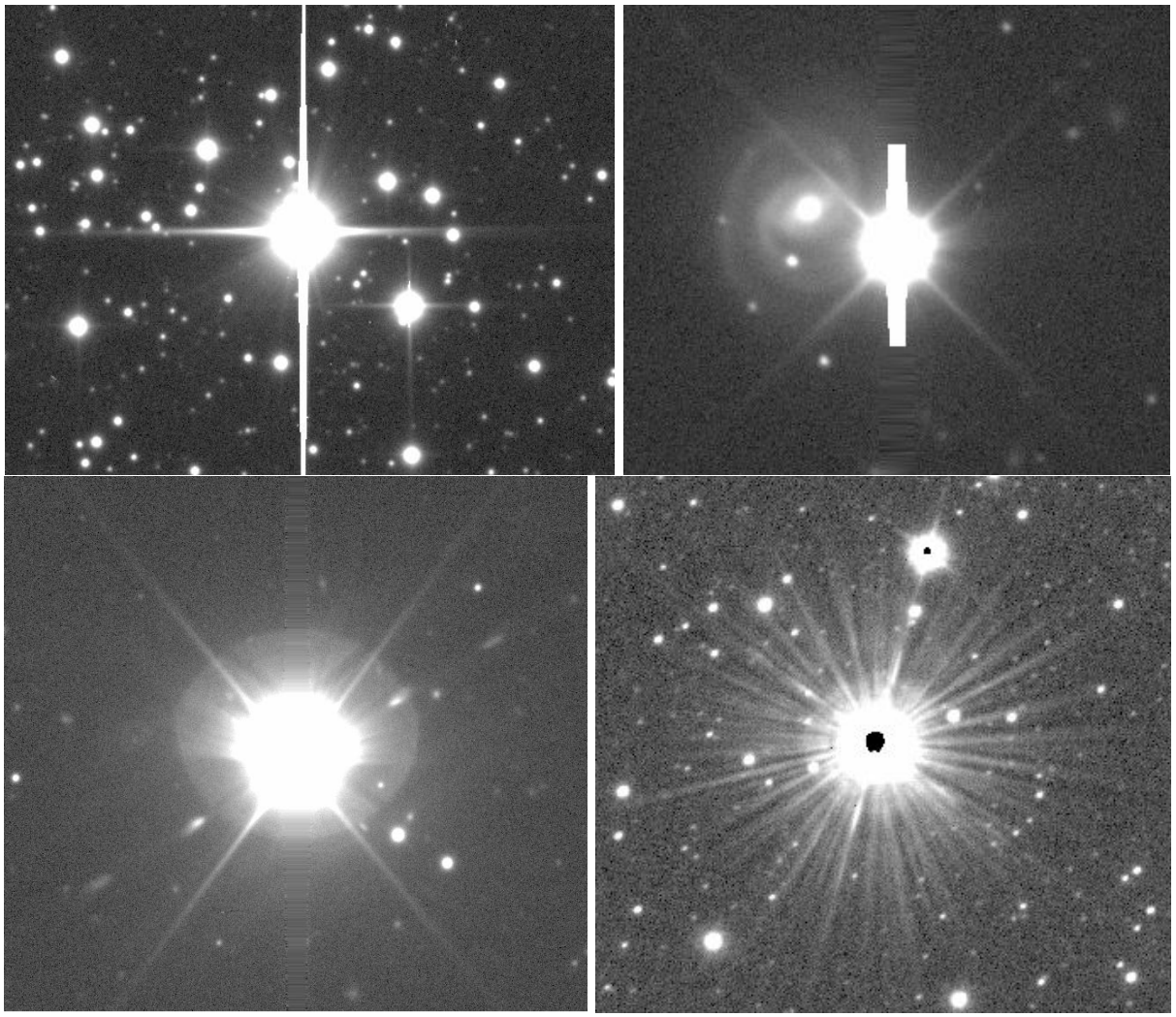


Figure 5.7: Examples of diffraction spikes. Top left: Megacam, an instrument where spikes are '+'-shaped and thus overlap with bleeding saturation patterns. Top right: DECam, an instrument with 'x'-shaped spikes that sometimes also exhibits an additional horizontal spikes. Bottom left: HSC, an instrument where the spike pattern can vary a lot from an exposure to another but also within the focal plane. Note also the small clumpy *spikes* along the diagonal. Bottom right: VISTA, an instrument where multiple spikes appear during exposures. Note the different saturation pattern as it is an infrared camera.

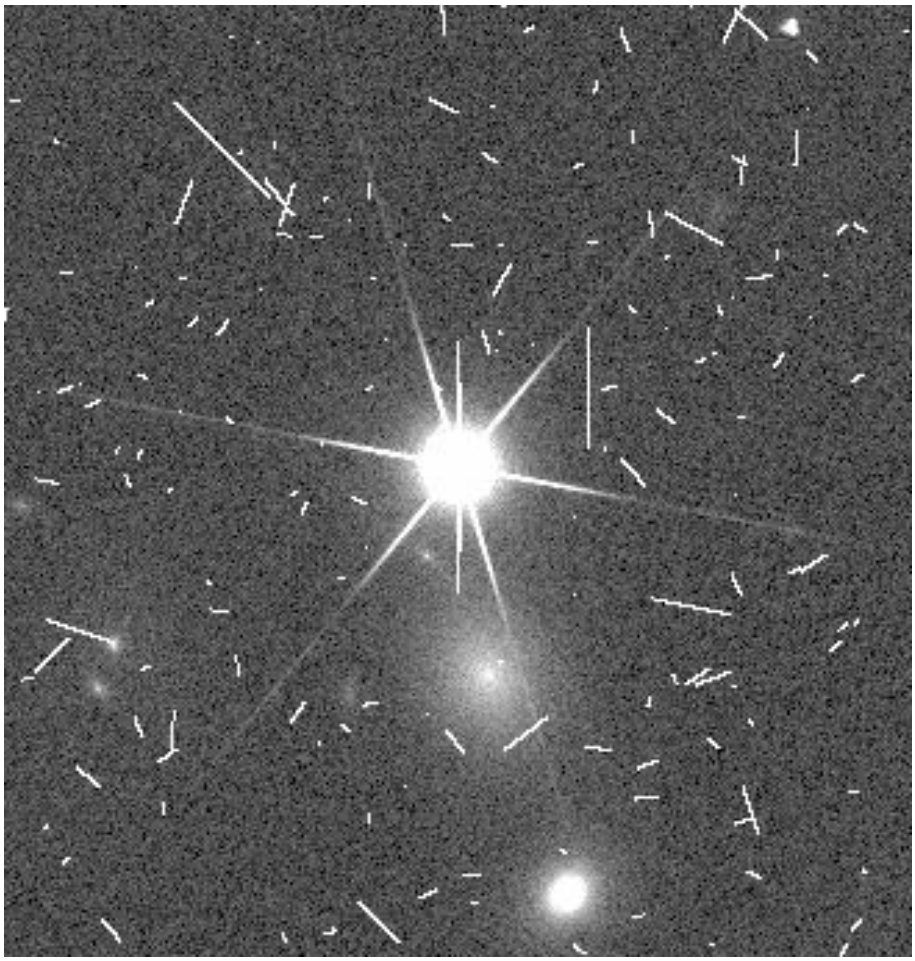


Figure 5.8: Example of diffraction spikes in a Euclid simulated image. Image credit: Euclid simulation group (IAP).

Star halos and ghosts

Star halos, also sometimes called "ghosts", are local contaminants produced by bright stars in or near the focal plane. Very bright stars do not only produce bleeding trails and diffraction spikes but also halos, which are images of the pupil of the telescope produced because of the light of the bright star. Their position and focusing depend on the inclination of the light-rays and the brightness of the star. Several halos may be present, they may be strongly defocused, and located far away from the bright star that causes them (Fig. 5.9).

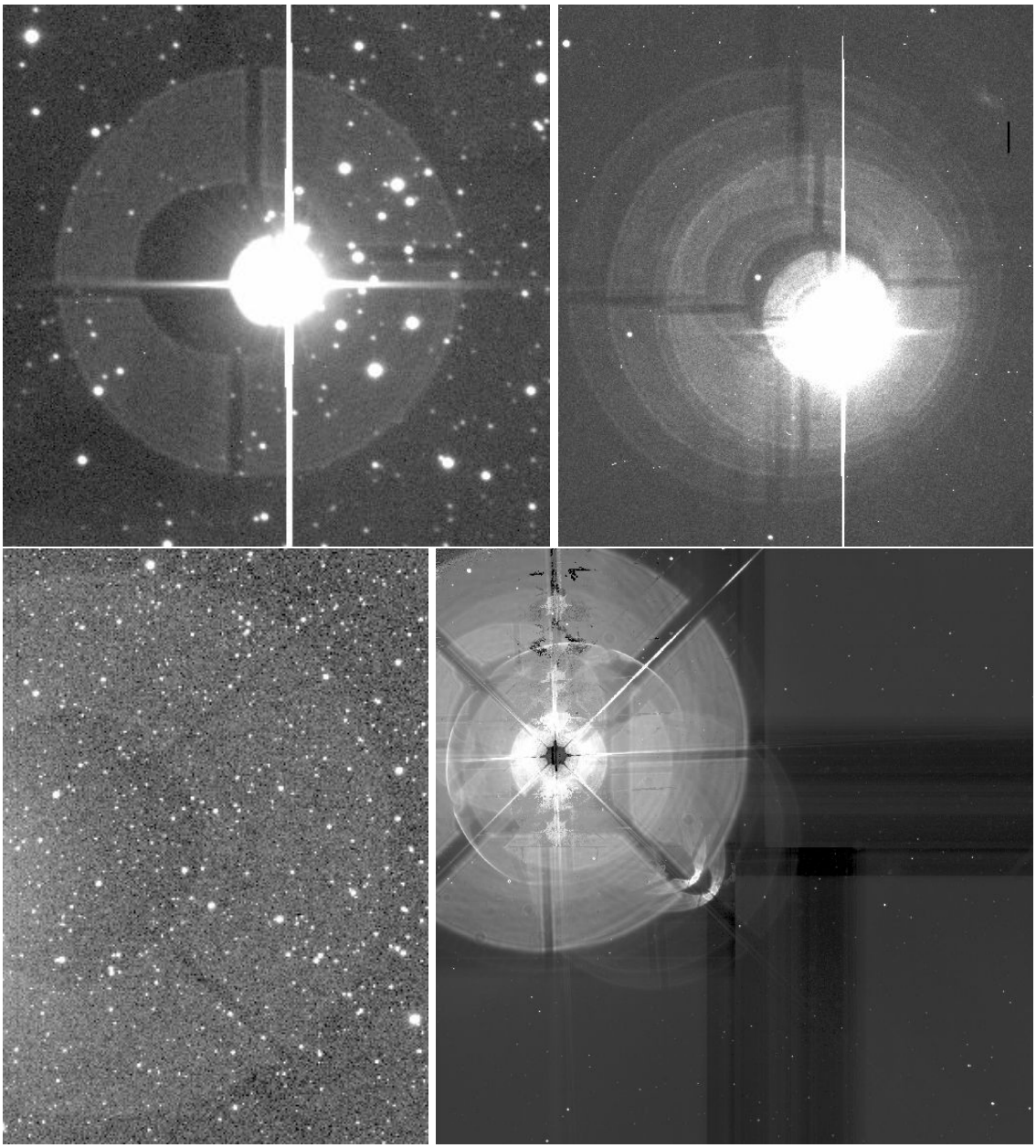


Figure 5.9: Examples of star halos and ghosts. Top: two examples from Megacam images. Note how the left halo is defocused and how the right example exhibits several defocused halos. Bottom left: an example from DECam where the resulting ghost occurs in a different CCD than the bright star causing it. Bottom right: an example from WFCAM.

Reflections, flares and scattered light

Undesired reflections in the telescope can result in local contaminants, including flares and scattered light patterns in images. They are even more prevalent in wide-field instruments which often use complex combinations of coated lenses to correct for field aberrations. Scattered light can also occur when a bright star is outside the field of view but close enough that some rays enter the telescope through series of reflections on its structure. Examples of flares are shown in Fig. 5.10.

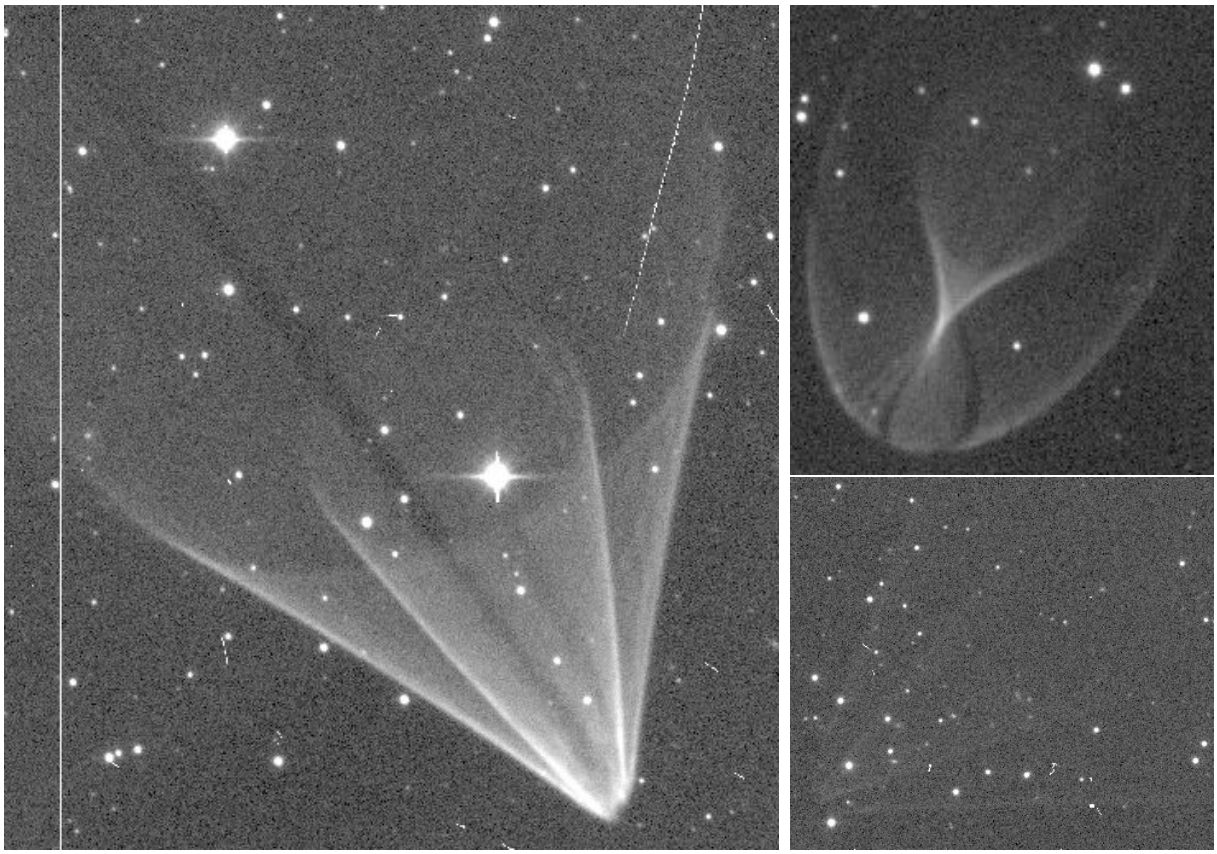


Figure 5.10: Examples of reflections in HSC. Left: a reflection pattern spreading over a large part of the CCD. Top right: another reflection pattern. Bottom right: a fainter reflection pattern. These are related to the numerous lenses present in the HSC wide-field corrector.

5.1.3 Contaminants due to external events

The following contaminants are signals related to external physical processes not linked to the instrument itself.

Cosmic rays

Cosmic-ray hits are local contaminants related to high energy particles. They appear as bright and sharp patterns in images. The patterns can be almost point-like, straight lines or curved lines depending on their incidence angle with the detector (Fig. 5.11). They actually often result from the decay of radioactive atoms in materials near the detector, for example anti-reflection coatings.

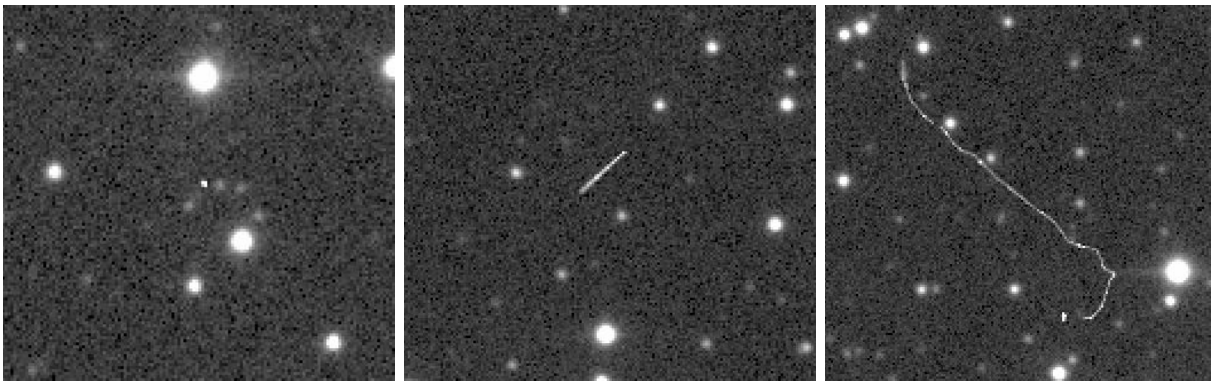


Figure 5.11: Examples of cosmic-ray hits in DECam exposures. Left: a point-like cosmic ray hit. Middle: a straight cosmic-ray track. Right: a worm-like cosmic ray feature (a speck-like impact is also visible).

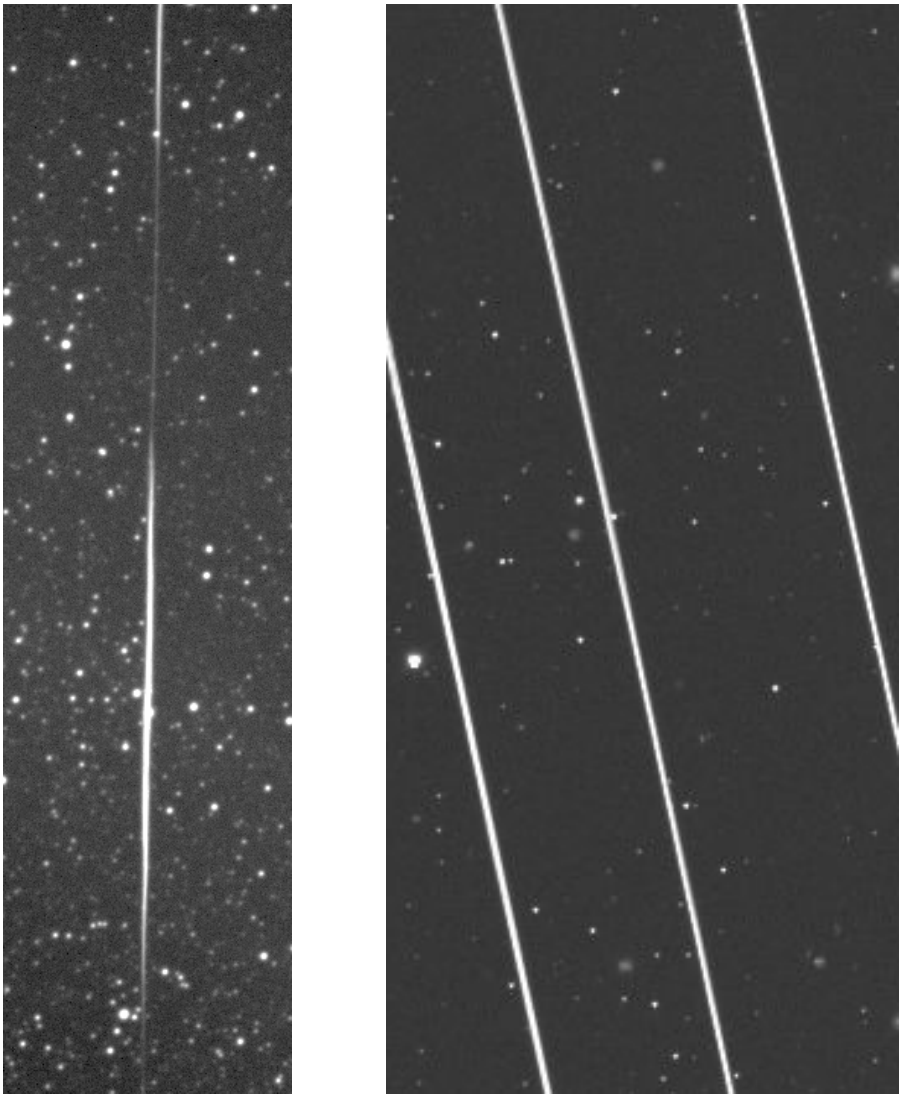


Figure 5.12: Left: example of a blinking trail in a ZTF survey image (https://www.ztf.caltech.edu/). Right: example of multiple trails in DECam images (train of Starlink satellites).

Trails

Trails are local contaminants caused by meteors, satellites or planes crossing the field of view during an exposure. They appear as long rectilinear trails in images. Trails due to plane or satellites are sometimes discontinuous or variable in amplitude (Fig. 5.12).

Nebulosities

From the astrophysical point of view, nebulosities are not contaminant but genuine astrophysical objects. Regarding the main objectives of the COSMIC-DANCE and Euclid surveys (as well as many other surveys interested in stars and galaxies), nebulosities are nevertheless a major source of nuisance and in this work we consider them as contaminants that greatly complicate or bias the source detection and measurement processes. Examples of nebulosities in images are shown in Fig. 5.13.

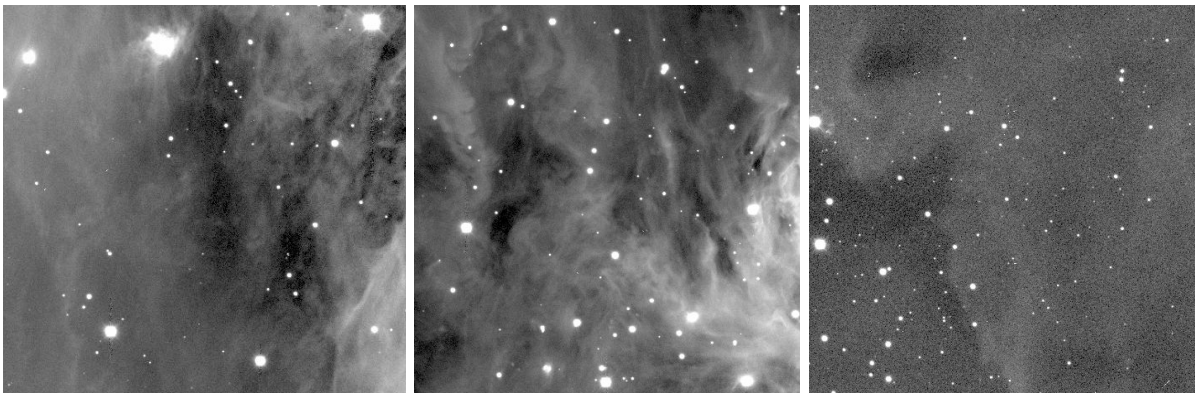


Figure 5.13: Three examples of nebulosities in DECam images.

5.1.4 Global contaminants

In addition to the local contaminants mentioned above there are also global contaminants affecting the entire image rather than a fraction of its pixels.

Tracking errors

The analysis of hundreds of thousands of archival and private exposures by the COSMIC-DANCE survey showed that telescope tracking or guiding errors happen from time to time, either because of a hardware/software failure, earthquakes (many observatories are located in seismically active areas, e.g. Hawaii, La Palma or the Chilean Andes), or wind gusts. Tracking/guiding errors result in images where all the sources are blurred and elongated along the telescope motion direction. Examples of images suffering such tracking errors are presented in Fig. 5.14.

Note that the non-sidereal tracking used for solar-system observations produces the same effect and cannot be distinguished from the above mentioned problems. Given that neither the COSMIC-DANCE or Euclid surveys are interested in this type of observations, we do not attempt to distinguish them from truly problematic images and include them in the tracking error category.

Figure 5.14: Left: tracking error in a CTIO-Mosaic2 image. Right: tracking error in a VISTA image.

Defocusing

Defocusing occurs when the detector is not perfectly positioned at the focal plane. It can happen either because the position of the focus has changed due to variations of ambient conditions, because of a human or software error while setting the focus, or purposely in some specific scientific cases. Examples of defocused images are shown in Fig. 5.15.

Figure 5.15: Left: defocusing in a KPNO-Mosaic1 exposure. Right: defocusing in a VIMOS exposure.

5.2 Identifying contaminants

Addressing the problem

Up to recently some pipelines have used visual inspection (e.g., Erben et al., 2005; Heymans et al., 2012) to identify and mask contaminants. But the amount and rate of astronomical data produced in current and upcoming surveys such as LSST (Ivezić et al., 2019) and Euclid (Racca et al., 2016) make visual inspection absolutely impossible. It is therefore crucial to develop fully automatic methods to detect the contaminants and separate them from the true astrophysical sources.

Our original motivation came from the COSMIC-DANCE survey (Bouy et al., 2013). It quickly became clear that contaminants were a major source of errors for the astrometric and photometric analysis of the tens of thousands of images obtained at many observatories and with many cameras. Detecting contaminants was particularly important in the case of archival images which had been obtained by other persons with observing strategies sometimes incompatible with the COSMIC-DANCE requirements.

The variety of the (non-exhaustive) list of contaminants given in Section 5.1 led most existing approaches to either:

- Be focused on a specific contaminant.
- Be tailored to a specific instrument by relying on prior knowledge of its properties.

Let me now discuss both approaches and come up with a new proposal.

Modern pipelines

A classic observing strategy employed in modern survey consists in taking multiple exposures of the same field. A small offset is usually applied between each of these exposures. This method, called "dithering", is particularly useful to detect transient defects that are likely to affect only one of the individual images such as cosmic rays or trails. By comparing the individual images with the stacked image, one can identify these types of defects. This is one of the solutions chosen for the LSST (Bosch et al., 2019). Methods developed in that sense also include Gruen et al. (2014) and Desai et al. (2016). The drizzle algorithm (Fruchter and Hook, 2002) originally designed to improve the sampling of under-sampled dithered images also provides means to manage cosmic rays by taking advantage of the timeline.

Yet, not all the surveys adopt this strategy or simply cannot afford to take multiple exposures of the same field. This is why modern pipelines also rely on a strong prior knowledge of their instruments and are finely tuned for their images. For example, the HSC pipeline (Bosch et al., 2018) and the DECam pipeline (Morganson et al., 2018) detect and mask electronic contaminants and to some extent optical contaminants (Kawanomoto et al., 2016a,b). Using such knowledge makes them very efficient on their data but the downside is that the analysis becomes instrument dependent and cannot be directly applied to other data.

Methods focusing on a specific contaminant

Some methods chose to be more universal (in the instrumental sense) but focus on detecting specifically a given type of contaminant. Cosmic-ray detection algorithms are probably the most illustrative example of this class of software tools. They include simple linear filtering and thresholding (Rhoads, 2000) to identify cosmic ray impacts in well sampled images, Laplacian edge detection in LACosmic (van Dokkum, 2001) or bright outlier search in histograms (Pych, 2004). Farage and Pimbblet (2005) propose a comparison of these techniques. The method

implemented in LACosmic has become very popular in spite of its sluggishness (van Dokkum et al., 2012), even in its more modern and optimized Python version astroscrappy (McCully and Tewes, 2019).

Some other methods have been designed to identify trails, that are becoming more and more numerous in images. Algorithms based on the Hough transform (Hough, 1962) to detect lines have been developed (Cheselka, 1999; Vandame, 2002; Storkey et al., 2004; Bektešević and Vinković, 2017). In most of these approaches, the Hough transform is also applied to detect other trail-like features such as bad columns or diffraction spikes. The Radon transform (Radon, 1917, 1986) has also been used to detect trails (Nir et al., 2018).

Finally, some multiscale methods using wavelets are used (Ordénovic et al., 2008) to detect glitches.

Our strategy

Following our original goal of designing a universal (meaning not instrument-specific), unsupervised and robust source detector, we want to overcome the drawbacks of the above presented methods. Several reasons mentioned in Section 3.4 led us to chose a data driven approach instead of more classical algorithmic approaches. The main motivations include:

- The data volume produced by COSMIC-DANCE and Euclid as well as other modern surveys that leave no other choice than to develop largely automatic and unsupervised tools.
- The superior efficiency of recent data-driven approaches in various computer vision tasks.
- The need for a generic tool that could ideally detect all contaminants at once and for a wide range of instruments without any instrument-specific input parameters.

All these reasons led us to experiment with CNNs. Neural networks and CNNs have already been introduced in Chapter 4 for image classification. Here we do not aim at performing image classification but we want to know whether and where contaminants are present in an image at the pixel level. To do so, CNNs are modified to output prediction maps at the same resolution as the input image instead of predictions at the image level. The corresponding task of classifying pixels is called semantic segmentation. We must now introduce some concepts complementary to those introduced in Chapter 4 to extend the use of CNNs to semantic segmentation.

5.3 CNNs for semantic segmentation

Let us start with a brief review of CNNs for semantic segmentation, focusing only on fully convolutional networks. For a broader and more exhaustive review of semantic segmentation deep learning techniques, I refer the reader to Garcia-Garcia et al. (2017).

5.3.1 Fully convolutional neural networks for semantic segmentation

The main difference between CNN classifiers and semantic segmentation CNNs is the ability of the latter to recover predictions at the same spatial resolution as that of the input image. To achieve this, a new type of layer is introduced to upsample the feature maps, i.e., to increase their spatial resolution back to the initial spatial resolution. In a classical CNN for image classification, several convolution layers are stacked and the resulting feature maps are fetched into fully connected layers for classification. In a fully convolutional neural network, there are no fully connected layers: instead, there are stacks of upsampling layers, each one corresponding

to a subsampling layer of the first part of the CNN. Doing so, the feature maps from the last convolution layer are progressively upsampled to recover the initial image spatial resolution.

Basic upsampling

There are several ways to construct upsampling layers. The most basic one is to directly increase the size of the feature maps and interpolate the missing values, usually with nearest neighbor or bilinear interpolations. But it is far from ideal because the upsampling operation is fixed and not learnable by the CNN. Ideally, one would want to learn the upsampling operation by introducing learnable parameters for upsampling. This is why other upsampling techniques have been preferred.

Upsampling with transposed convolutions

It is possible to learn the upsampling layer thanks to the deconvolution layer, which is not a very appropriate name as it does not perform a deconvolution as defined in signal theory. Instead, the operation should be more wisely named "transposed convolution" (it is also known as fractional convolutional layer and up- or backward-convolutional layer). The transposed convolution consists of swapping the forward and backward passes of a regular convolution. A good resource explaining transposed convolutions is Dumoulin and Visin (2016)[3]. Even if in practice it is not implemented as such, transposed convolution can be seen as a regular convolution applied to a padded input, as shown in Fig. 5.16.

Long et al. (2015) were among the first to introduce such upsampling layers to make dense predictions with a fully convolutional network. It was later improved in the U-net architecture (Ronneberger et al., 2015), which remains one of the main references in semantic segmentation. A U-net exhibits the typical semantic segmentation network architecture, and is divided in two parts. The first part uses classical convolution and pooling layers and progressively decreases the spatial resolution of the feature maps. The second one uses upsampling and convolution layers and progressively increases the spatial resolution of the feature maps. Each layer in the first part has a corresponding layer in the second part, hence recovering the initial spatial resolution in the last layer. It also uses skip connections: the feature maps of the first part of the network are concatenated with the upsampled feature maps of the second part of the network which have the same spatial resolution. It makes it possible to use the maximum of information present in the network and to have a better error gradient propagation to the first layers. The overall architecture is shown in Fig. 5.17.

[3] https://github.com/vdumoulin/conv_arithmetic

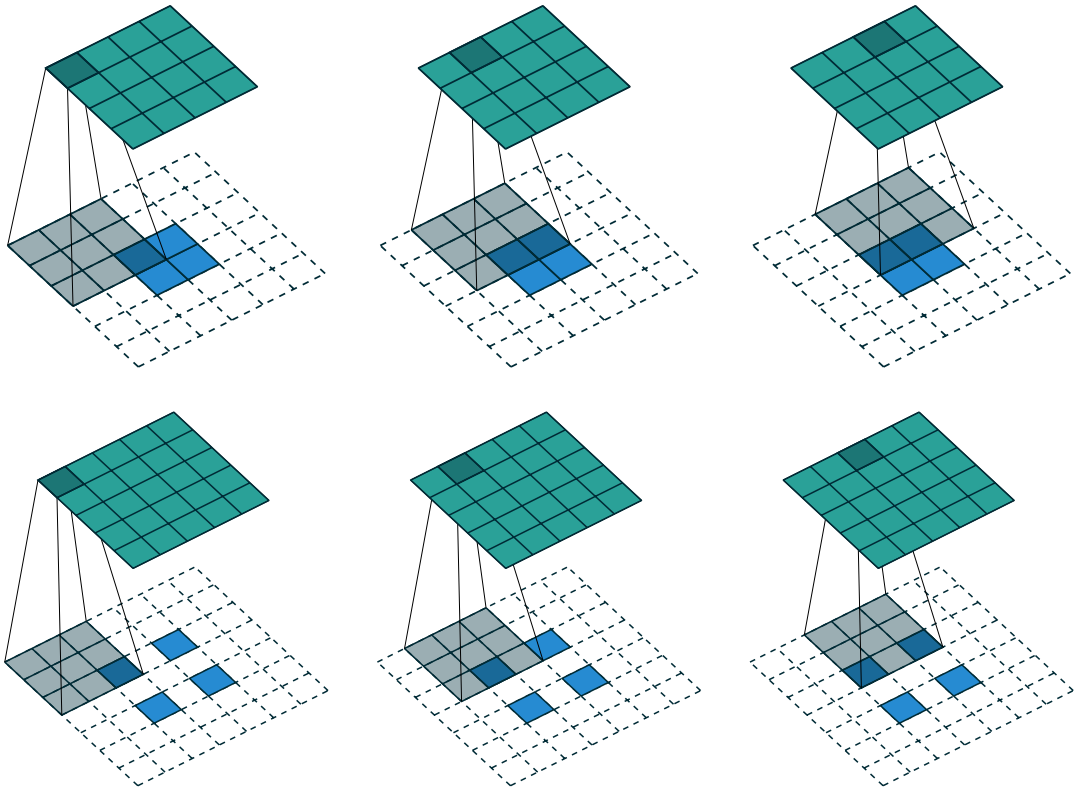


Figure 5.16: Top: tranposed convolution with stride one. Bottom: tranposed convolution with stride two. The operations are equivalent to regular convolutions with appropriate padding. Images credit: Dumoulin and Visin (2016).

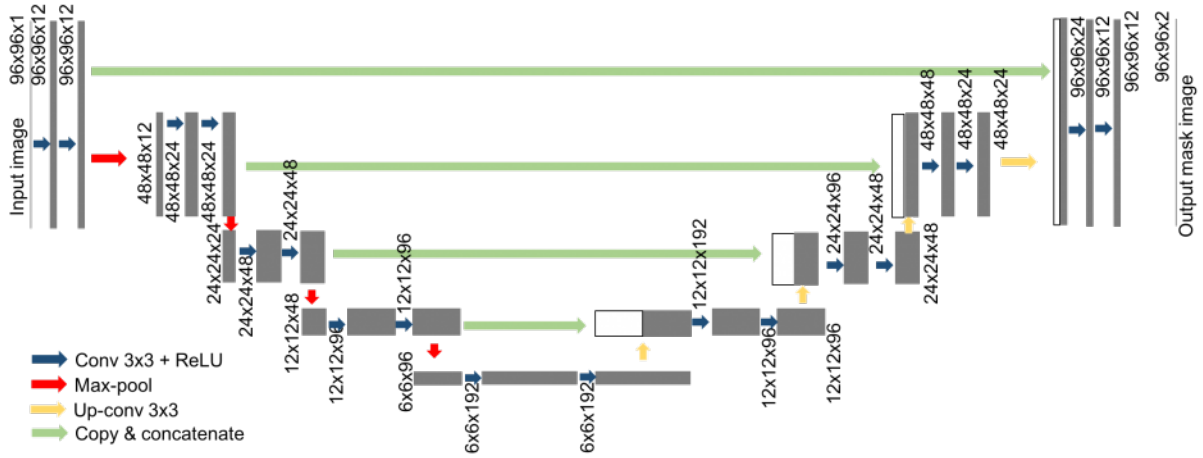


Figure 5.17: U-NET-like architecture with four resolution levels. Each convolution layer of the first part of the network has a corresponding upsampling layer in the second part. There are also skip connections between the same spatial resolution feature maps. Image credit: https://www.depends-on-the-definition.com/about/.

Upsampling with unpooling

Finally, another upsampling technique is unpooling. The feature maps of the second part of the network are upsampled using the recorded max pooling indices from the corresponding feature maps of the first part of the network. The feature-map values are simply placed at the corresponding locations in the higher resolution feature map, as shown in Fig. 5.18.

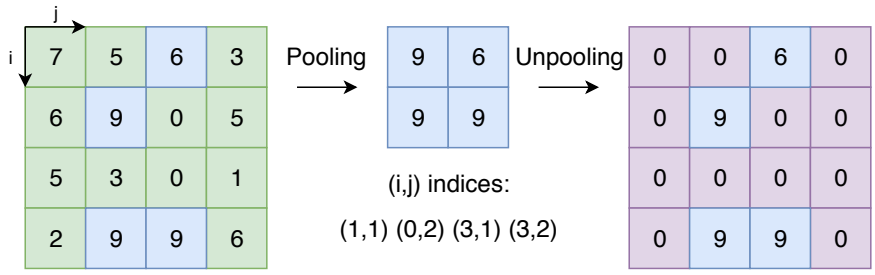


Figure 5.18: Unpooling operation. The pooling indices at a given spatial resolution are stored to upsample to this resolution in the second part of the network.

Although it has already been used in Zeiler et al. (2011) and Zeiler and Fergus (2014), unpooling has been popularized in semantic segmentation by SegNet (Badrinarayanan et al., 2015, 2017), which has also become a classical semantic segmentation network architecture (Fig. 5.19). SegNet is very similar to U-net, it but does not use the same upsampling layers and does not use skip connections. Both architectures are often used with the VGG (Simonyan and Zisserman, 2014) backbone architecture presented in Section 4.3.3.

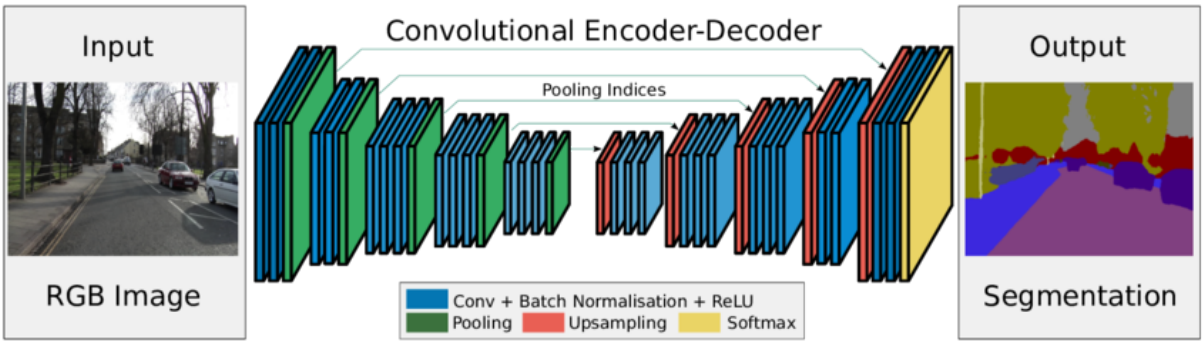


Figure 5.19: SegNet architecture. Max pooling indices from the first part of the CNN are reused in the second part for upsampling the feature maps to higher spatial resolutions. Image credit: Badrinarayanan et al. (2015, 2017).

The loss functions used to train these models are similar to the commonly used loss functions for classification discussed in Section 4.2.4, except that those are applied at the pixel level: each classified instance is a pixel and not an image.

5.3.2 Applying CNNs to the identification of astronomical contaminants

CNN-based methods have rarely been used for the identification of astronomical contaminants so far. Besides the "venerable" EyE package already mentioned in section 4.3.2, only a handful of tools were available at the time of writing this thesis. Only one performs semantic segmentation, and none deals with more than one type of contaminants.

In addition to cosmic-ray detection in HST images, DEEPCR (Zhang and Bloom, 2020) can do inpainting, i.e., replacing pixels hit by a cosmic-ray with the expected uncontaminated values. DEEPCR is however restricted to cosmic rays and tailored to HST data so that it may not adapt well to other instruments. Finally, it is also likely to trigger false detections on other artifacts like bleeding trails or hot columns.

Other approaches have used CNNs for image classification in order to predict the presence of contaminants without performing semantic segmentation. Paranjpye et al. (2020) perform image artifact classification using a CNN and Teimoorinia et al. (2020) detect tracking errors and bad seeing images with a hybrid method using self organizing maps (Kohonen, 1982) and a CNN.

5.3.3 MaxiMask and MaxiTrack

In view of the above, we must address the issues of local and global contaminants separately. The identification of local contaminants will be based on semantic segmentation (MaxiMask), while that of global contaminants will rely on image classification (MaxiTrack), which will require different CNN architectures.

5.4 Data sets

In order to train our CNNs, we prepare our own data samples. Independently from the local or global nature of contaminants, we aim to use real data as much as possible to maximize the inference capabilities of MaxiMask and MaxiTrack with real data. Whenever real data is gathered to build learning samples, we reserve 75% of the data to build training samples and 25% to build testing samples.

5.4.1 Overview of the data

We mainly use data from the COSMIC-DANCE (Bouy et al., 2013) wide-field private archives. A list of the COSMIC-DANCE instruments used in this work is presented in Table 5.1. They include modern and first generation CCD and NIR detectors encompassing 20 years of technological development.

Images from all instruments are reduced with an updated version of Alambic (Vandame, 2002), except for:

- Megacam images that are reduced with the Elixir pipeline (Magnier and Cuillandre, 2004).
- DECam images that are reduced with the DECam community pipeline (Valdes et al., 2014).
- UKIRT images that are reduced by the Cambridge Astronomical Survey Unit and retrieved from the WFCAM Science archive.
- HSC images that are reduced by the HSC pipeline (Bosch et al., 2018).

In order to build the MaxiMask and MaxiTrack training samples, we adopt the following strategies:

- MaxiMask: we build training samples by adding contaminants to uncontaminated images. As none of our images comes perfectly uncontaminated, we identify the *cleanest* images among our data. We opt for images from the CFHT-Megacam, CTIO-DECam and Subaru-HSC instruments as a basis for uncontaminated images. We use the instrument's pipelines to identify the main artifacts like cosmic rays and bad pixels, which are replaced using

Telescope	Instrument	Type	Platescale [pixel^{-1}]	Ref.
CTIO Blanco	DECam	CCD	0.″26	(1)
CTIO Blanco	MOSAIC2	CCD	0.″26	(2)
KPNO Mayall	MOSAIC1	CCD	0.″26	(2)
KPNO Mayall	NEWFIRM	IR	0.″40	(3)
CFHT	Megacam	CCD	0.″18	(4)
CFHT	CFH12K	CCD	0.″21	(5)
CFHT	UH8K	CCD	0.″21	(6)
INT	WFC	CCD	0.″33	(7)
UKIRT	WFCAM	IR	0.″40	(8)
LCO Swope	Direct CCD	CCD	0.″43	(9)
VST	OmegaCam	CCD	0.″21	(10)
Subaru	HSC	CCD	0.″17	(11)
VISTA	VIRCAM	IR	0.″34	(12)

Table 5.1: Imaging instruments used from the COSMIC-DANCE survey. References: (1) Flaugher et al. (2010) ; (2) Wolfe et al. (2000) ; (3) Autry et al. (2003) ; (4) Boulade et al. (2003) ; (5) Cuillandre et al. (2000) ; (6) Metzger et al. (1995) ; (7) Ives (1998) ; (8) Casali et al. (2007) ; (9) Rheault et al. (2014) ; (10) Kuijken et al. (2002) ; (11) Miyazaki et al. (2018) ; (12) Dalton et al. (2006). Table credit: Paillassa et al. (2020)

Gaussian interpolation (Williams, 1998). Additionally, we use LACosmic (van Dokkum, 2001; McCully and Tewes, 2019) to identify cosmic rays that would not have been detected by the instrument's pipelines. The contaminant addition procedures are explained in more details in the next section.

- MaxiTrack: we build training samples by simply gathering the images visually identified as affected by tracking errors through the years. These visual inspections were mainly made in the context of the COSMIC-DANCE survey and by Mike Read at the UKIRT telescope. For now MaxiTrack remains limited to identifying tracking errors and does not include the identification of defocused images as not enough defocused images could be gathered. Note that as the main use of MaxiTrack is to detect images affected by tracking errors prior to any further data processing, it must be able to work with images affected or not by local contaminants. Thus, the MaxiTrack training should ideally contain both images affected by local contaminants and images that are not.

The list of archive images from the COSMIC-DANCE survey used to build our training data sets is presented in Table 5.2, while an overview of the training sample production is shown in Fig. 5.20.

Instrument	Clean	CR	No TR	TR
DECam	✓		✓	
MOSAIC2		✓		
MOSAIC1		✓		
NEWFIRM			✓	✓
Megacam	✓	✓	✓	✓
CFH12K		✓	✓	✓
CFH8K				✓
WFC			✓	✓
WFCAM			✓	✓
Direct CCD (LCO Swope)			✓	✓
VST		✓	✓	✓
HSC	✓	✓	✓	
VIRCAM			✓	✓

Table 5.2: COSMIC-DANCE archive usage per imaging instrument: *Clean* is for uncontaminated images, *CR* for dark images used for cosmic-ray addition, *No TR* is for images *not* affected by tracking errors, and *TR* for images affected by tracking errors. Table credit: Paillassa et al. (2020).

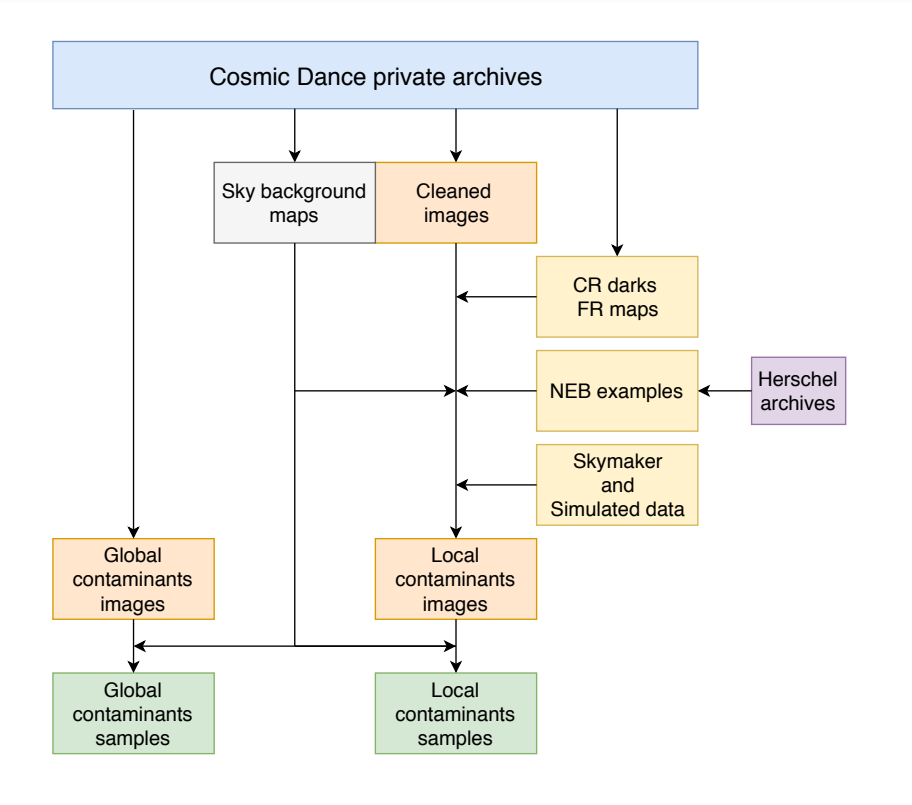


Figure 5.20: Overview of the sample generation pipeline. See Table 5.3 for translations of the acronyms. Image credit: Paillassa et al. (2020).

Note that the sky background of the images and its standard deviation are estimated. This is made using a method similar to SExtractor's (Bertin and Arnouts, 1996), i.e., using k-σ clipping and mode estimations. It serves two purposes. Firstly, the standard deviation of the

sky-background is widely used in the contaminant-addition procedures described in Section 5.4.2. Secondly, background subtraction is part of the image preprocessing (Section 5.4.2).

5.4.2 MaxiMask training samples

As mentioned in Section 5.4.1, the strategy to build the MaxiMask training samples is to contaminate uncontaminated images. In the same way that we favor real data to gather uncontaminated images, we also aim to retrieve the contaminants from real data as much as possible. A list of all the contaminants currently included in MaxiMask is presented in Tab. 5.3, along with their provenance.

Contaminant	Abbreviation	Data origin
Cosmic rays	CR	Real: dark images
Hot columns/lines	HCL	Simulations
Dead columns/lines/clusters	DCL	Simulations
Hot pixels	HP	Simulations
Dead pixels	DP	Simulations
Persistence	P	Simulations from model (1) and SkyMaker (2)
Trails	TRL	Simulations with SkyMaker (2)
Fringes	FR	Real: fringing maps
Nebulosities	NEB	Real: Herschel SPIRE (3) (4)
Saturated pixels	SAT	Inherent
Diffraction spikes	SP	Inherent
Overscanned pixels	OV	Simulations
Bright background	BBG	Inherent
Background	BG	Inherent

Table 5.3: List of all the contaminants along with their abbreviated names and the origin of the data. The "inherent" origin means that the contaminant is naturally present in the images. References: (1) Long et al. (2015) (2) Bertin (2009) (3) (Pilbratt et al., 2010) (4) (Griffin et al., 2010). Table credit: Paillassa et al. (2020).

Most contaminants are added directly in the uncontaminated images given an adequate scaling. Yet, the contaminants that are marked "inherent" in Table 5.3 are already present in the uncontaminated images. Therefore, they must be identified within the uncontaminated image, before the addition of any contaminant. This is the case for saturated pixels, diffraction spikes and "bright backgrounds". The latter contains the astrophysical objects already present in the image. This category was created with the sole purpose of improving the training performance (which it did).

Added contaminants

The added contaminants are almost all scaled in the uncontaminated image with respect to the sky-background standard deviations of the images. In the following equations, the uncontaminated image is noted $\boldsymbol{U}$, the contaminated image is noted $\boldsymbol{C}$ and the standard deviation of the uncontaminated image sky background is noted $\sigma_{\boldsymbol{U}}$.

Cosmic ray hits: In order to add realistic cosmic ray hits, we extract them from dark images from the CFH12K, HSC, Megacam, MOSAIC and OmegaCam cameras (Table 5.2). These instruments include both thick, red-sensitive CCDs and thin, blue-sensitive CCDs, providing

instances of all the types of cosmic rays described in Section 5.1.3. As dark images are taken with a closed shutter, only the contribution of the offset, the dark current and cosmic-ray hits remain in the images, as well as Poisson noise and Gaussian readout noise. This allows us to easily identify the cosmic rays using thresholding: the dark image $\boldsymbol{D}$ is background-subtracted and we set a threshold to $3\sigma_D$ above the median value m_D of the dark image:

$$\forall p,\ M_p = \begin{cases} 1 & \text{if } D_p > m_D + 3\sigma_D \\ 0 & \text{otherwise.} \end{cases} \tag{5.1}$$

Among all the masks computed from the dark data, we make a selection based on two criteria. Firstly, we reject masks where columns or lines contain too many pixels identified as cosmic rays to avoid getting false detections of hot columns, lines or other defect features. Secondly, we retain only the masks containing a minimum fraction of pixels identified as cosmic rays of 0.0002 to make sure to add a minimum number of cosmic rays in the uncontaminated images.

Within the retained dark images and masks, a bit more than 900 million cosmic-ray pixels are detected after thresholding. Considering that the average footprint area of a cosmic-ray hit is 15 pixels, this represents a richly diversified population of about 60 million cosmic-ray objects instances.

Then we dilate these masks with a 3×3 square structuring element. The resulting masks are used both to add the cosmic rays in the uncontaminated image and to be the ground truth masks in order to generously mask these defects. The addition to the uncontaminated image $\boldsymbol{U}$ is done by rescaling according to the standard deviations of the images (Fig. 5.21):

$$\boldsymbol{C} = \boldsymbol{U} + k_{\boldsymbol{C}} \frac{\sigma_{\boldsymbol{U}}}{\sigma_{\boldsymbol{D}}} \boldsymbol{D} \odot \boldsymbol{M}^{(\boldsymbol{D})}, \tag{5.2}$$

where $\sigma_{\boldsymbol{D}}$ is the standard deviation of the dark image background, $\boldsymbol{M}^{(\boldsymbol{D})}$ is the dilated cosmic-ray mask and $k_{\boldsymbol{C}}$ is an empirical scaling factor set to 1/8.

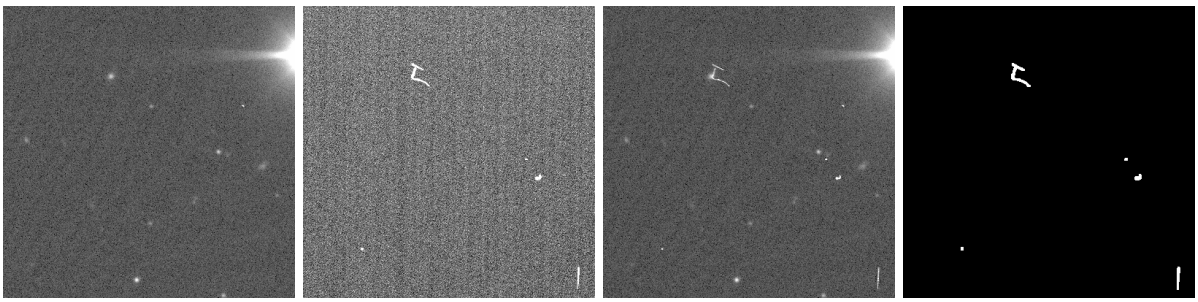


Figure 5.21: Example of cosmic-ray hits added to an HSC exposure. From left to right: the uncontaminated image, the dark image, the cosmic-ray contaminated image, the cosmic-ray ground-truth mask.

Hot and dead pixels: Both hot and dead pixels are simulated. We simulate different shapes of hot and bad pixels to be able to detect all the cases encountered in real images. These include columns, lines, point-like pixels for hot and dead pixels. We also simulate some small dead-pixel clumps and dead-pixel clusters (see Paillassa et al. (2020) for more technical details about all the simulations).

Even if all shapes of hot or dead pixels share a common origin, point-like defects are treated as a separate class by MaxiMask because or their distinctive pattern. This explains why there are two different ground-truth masks in the examples of hot and dead pixels of Fig. 5.22 and Fig. 5.23, respectively.

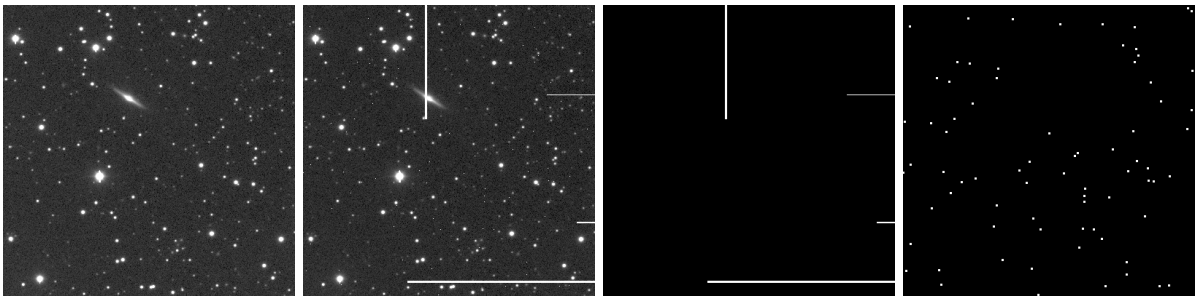


Figure 5.22: Examples of hot pixels added to a DECam exposure. From left to right: the uncontaminated image, the hot-pixel contaminated image, the hot column and line ground-truth mask, the point-like hot-pixel ground truth mask. The point-like hot-pixel ground-truth mask has been dilated for visualization.

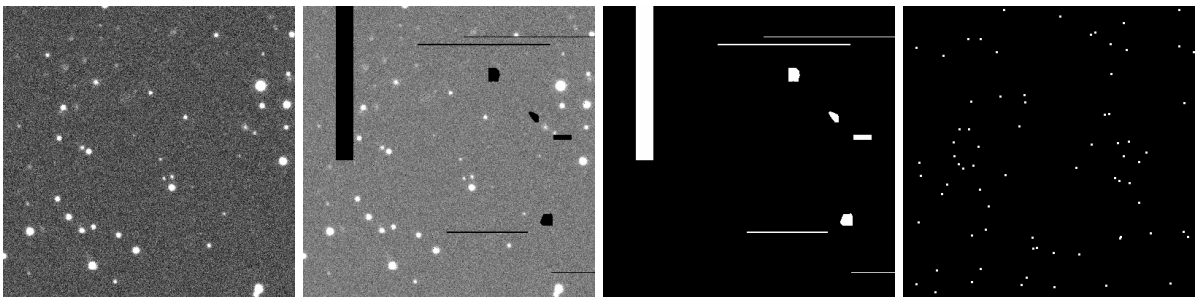


Figure 5.23: Examples of dead pixels added to a DECam image. From left to right: the uncontaminated image, the dead-pixel contaminated image, the dead column, line and cluster ground-truth mask, the point-like dead-pixel ground-truth mask. The point=like dead pixel ground truth mask is dilated for visualization.

Persistence: Persistence effects are simulated using the "Fermi model" developed by the STScI[4] for the HST WFC instrument. See Long et al. (2015) and Paillassa et al. (2020) for more technical details about the model. To simulate the footprints of the persistence patterns, we simulate saturated stars using SkyMaker (Bertin, 2009) that act as the virtual sources of a previous exposure causing the persistence effect. The simulations use the same pixel size and FWHM as the uncontaminated image. The FWHM retained for an image is the mean of the FWHM of the stars computed using PSFEx (Bertin, 2011). The saturated pixels of the stars define the persistence footprint as well as the ground-truth persistence mask. The pattern is added to the uncontaminated image according to:

$$\boldsymbol{C} = \boldsymbol{U} + k_{\boldsymbol{P}}\,\sigma_{\boldsymbol{U}}\frac{\boldsymbol{P} - P_{min}}{(P_{max} - P_{min})}, \tag{5.3}$$

where $\boldsymbol{P}$ are the persistence values computed with the "Fermi model" (Long et al., 2015; Paillassa et al., 2020), P_{min} and P_{max} are the minimum and maximum of these values and $k_{\boldsymbol{P}}$ is a scaling factor empirically set to 5. An example of a simulated remnant is shown in Fig. 5.24.

[4] https://www.stsci.edu/hst/instrumentation/wfc3/data-analysis/ir-persistence

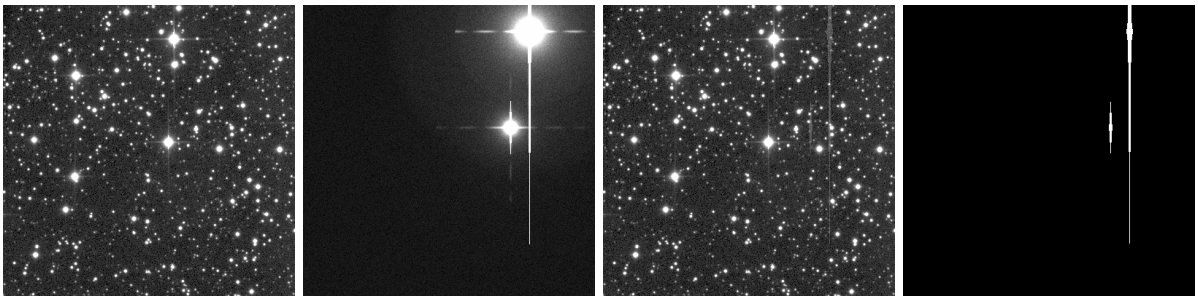


Figure 5.24: Example of persistence effects added to a Megacam image. From left to right: the uncontaminated image, the simulated saturated stars, the contaminated image, the persistence ground-truth mask.

Trails: Trails are simulated as motion-blurred artifacts using SkyMaker: close stars with identical magnitudes are simulated along a line with a small amount of Gaussian noise added to the source's positions to simulate jittering from atmospheric turbulence. Three types of trails are simulated:

- Trails exhibiting a constant brightness.
- Trails exhibiting brightness variation with a linear transition between the different brightness parts.
- Trails simulating close objects and thus exhibiting defocusing. The amount of defocusing θ, expressed as the apparent width of the pupil pattern in arc-seconds, is given by:

$$\theta = \frac{180}{\pi} \times 3600 \times \frac{D}{d}, \tag{5.4}$$

  where D is the diameter of the primary mirror and d is the distance between the object and the instrument, uniformly picked in $[2, 8]$ and $[80,000, 120,000]$, respectively, in meters.

Since the publication of the MaxiMask paper, there have been some additional work and improvements concerning trails, among which including fainter trails and making the ground-truth masks larger. For fainter trails, it is difficult to obtain a clean ground-truth mask by simply thresholding as described in Paillassa et al. (2020), even if the trail is isolated in the image. Thus, we now build the trail ground-truth mask from the positions of the sources that are simulated to form the trail: pixels corresponding to these source's positions are set to one and the resulting source-position map is dilated to build the trail ground-truth mask.

All types of trails are scaled with respect to their sky-background standard deviation $\sigma_{\boldsymbol{T}}$:

$$\boldsymbol{C} = \boldsymbol{U} + \frac{\sigma_{\boldsymbol{U}}}{\sigma_{\boldsymbol{T}}}\boldsymbol{T}. \tag{5.5}$$

An example of an added trail is shown in Fig. 5.25.

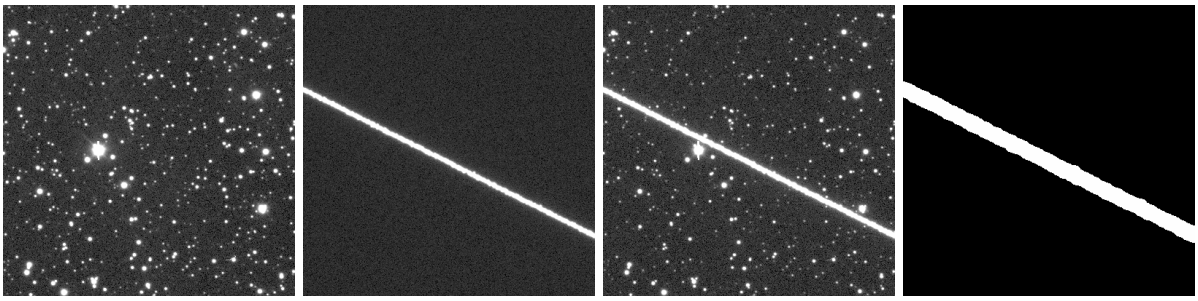


Figure 5.25: Example of a trail added to a DECam image. From left to right: the uncontaminated image, the simulated trail, the trail-contaminated image, the trail ground-truth mask.

Fringes: Residual fringing patterns are added from real fringing maps. We use fringing maps computed at the preprocessing level by the instrument's pipelines. As they are often affected by white noise, we smooth the maps with a top-hat kernel of size seven pixels. We mainly use fringing maps from the HSC instruments.

As fringing patterns and especially residual fringing patterns do not necessarily affect the whole image, we use a 3rd degree 2D-polynomial envelope to add fringing patterns only in some parts of the images. The envelope is rescaled over the interval $[-5, 5]$ and passed through a sigmoid function to draw a clear separation between the fringing and non fringing areas. The residual fringing pattern is then added to the image according to:

$$\boldsymbol{C} = \boldsymbol{U} + k_{\boldsymbol{F}} \frac{\sigma_{\boldsymbol{U}}}{\sigma_{\boldsymbol{F}}} \boldsymbol{F} \odot \boldsymbol{E}^{(\mathrm{F})}, \tag{5.6}$$

where $\boldsymbol{F}$ is the fringe image, $\boldsymbol{E}^{(\mathrm{F})}$ is the normalized sigmoid polynomial envelope, $\sigma_{\boldsymbol{F}}$ is the standard deviation of the fringe pattern and $k_{\boldsymbol{F}}$ is an empirical scaling factor set to 0.7. The ground-truth mask is obtained by thresholding the 2D-polynomial envelope to -0.025. An example of such a residual fringing pattern is shown in Fig. 5.26.

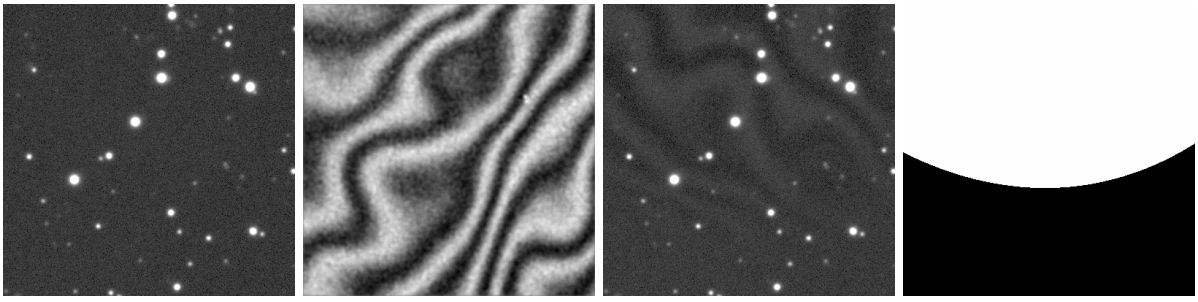


Figure 5.26: Example of a residual fringing pattern added to an HSC image. From left to right: the uncontaminated image, the smoothed fringing map, the fringe contaminated image, the fringe ground-truth mask.

Nebulosities: We choose as source of nebulosities far-infrared images of molecular clouds around star forming regions. More precisely, we retrieve from the science-archive pipeline the processed 250 μm images from the SPIRE instrument (Griffin et al., 2010) of the Herschel survey (Pilbratt et al., 2010). We make this choice for several reasons. Firstly, because the thermal distribution of dust corresponds well to the reflection nebulae at shorter wavelengths (Ienaka et al., 2013). Secondly, because there are mainly extended emission and few point sources at these low-latitude fields. Finally, because the 250μm channel offers the best compromise between signal-to-noise ratio and spatial resolution.

Taking advantage of the scale invariance of dust emission at the arcsecond level in molecular clouds (Miville-Deschênes et al., 2016), we do not resize or reconvolve the SPIRE images. The whole nebulosity image is background-subtracted using a SExtractor-like background estimation to form the final nebulosity pattern $\boldsymbol{N}$ which is then added to the uncontaminated image according to:

$$\boldsymbol{C} = \boldsymbol{U} + k_{\boldsymbol{N}} \frac{\sigma_{\boldsymbol{U}}}{\sigma_N} \boldsymbol{N}, \tag{5.7}$$

where σ_N the sky-background standard deviation of the nebulosity image and $k_{\boldsymbol{N}}$ is an empirical scaling factor set to 0.5. The ground-truth mask is computed by thresholding $\boldsymbol{N}$ above zero. This mask is then eroded with a 6-disk diameter structuring element to remove spurious individual pixels, and dilated with a 14-disk diameter structuring element. An example of nebulosity added to an image is shown in Fig. 5.27.

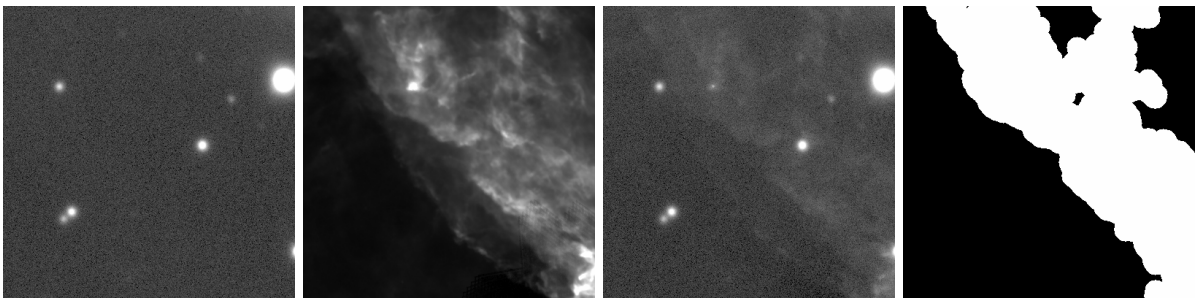


Figure 5.27: Example of a nebula added to an HSC image. From left to right: the uncontaminated image, the nebulosity image, the nebulosity-contaminated image, the nebulosity ground-truth mask.

Overscan: Overscan pixels are present in most CCDs. They are simply strips of very low pixel values at the borders of images. In order to avoid false detections in these regions, we include such features as an additional class. We simply simulate pixel strips with widths uniformly distributed between 15 and 50 pixels at the image borders (Fig. 5.28). Values follow the same rules as dead pixels.

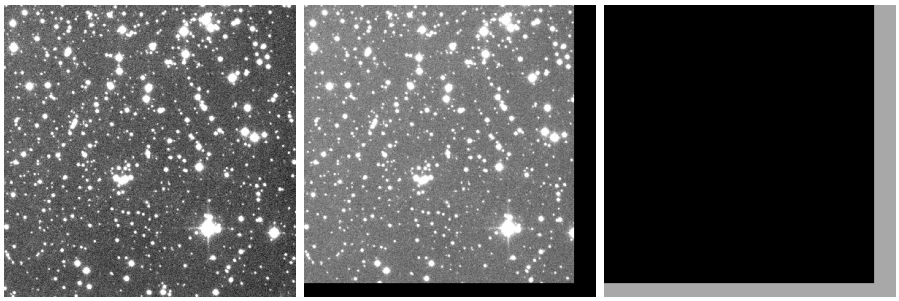


Figure 5.28: Example of an overscan added to a Megacam image. From left to right: the uncontaminated image, the simulated overscan, the overscan ground-truth mask. Overscan pixels are shown in gray for better visualization.

Inherent contaminants

As stated earlier, there are also contaminants that are "inherent" to images. These are directly identified in the images, before any other contaminant is added.

Saturation: Saturated pixels are simply identified in each uncontaminated image by using the known-saturation value of each instrument. Pixels above this limit are labeled as saturated and set to one in the saturation ground-truth mask (Fig. 5.29).

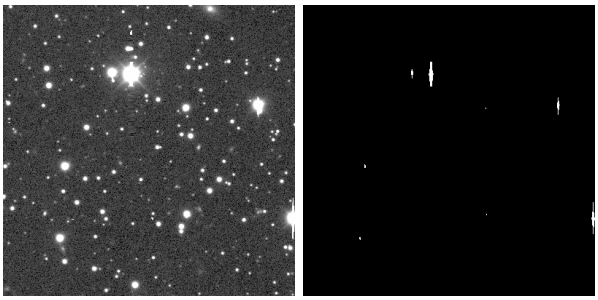


Figure 5.29: Examples of saturated pixels identified in a DECam image. Left: the input image. Right: the saturation ground-truth mask.

Diffraction spikes: As the Megacam and DECam instruments are mounted on Equatorial telescopes, they exhibit spike patterns that remain more or less identical across all images, consisting of '+' and 'x' cross shapes, respectively[5]. Unfortunately, it is not the case for HSC which is mounted on a telescope with an alt-azimuth mount, resulting in a diffraction spike pattern that can vary a lot across images. To identify the diffraction spikes for these three instruments, we use a two-step approach.

Firstly, we empirically identify diffraction spikes in Megacam and DECam images. To do so, we run SExtractor on the images to identify the brightest stars, i.e., the stars that are more likely to exhibit diffraction spikes, and extract 300×300 stamps centered on these bright stars. We threshold these stamps to 3σ above the sky background to obtain a first mask. We then compute the element-wise products of these masks with large centered '+' and 'x' patterns to isolate the potential diffraction spikes from the rest of the image. The resulting mask is matched-filtered with small horizontal and vertical line patterns and the result is thresholded to 15 ADU in order to remove eventual stars along the spikes. This final mask is used to compute an empirical diffraction-spike size. The length of each spike is measured from the center of the image to the borders as a contiguous block. To avoid measuring too large a size due to a neighboring star, the final spike size is taken as the maximum length found in all directions. If this length is too small, we consider it a false positive and no diffraction spikes are assigned to this star. An overview of the whole process is shown in Fig. 5.30.

Secondly, we use a U-net-like CNN architecture to identify the diffraction spikes in HSC images. To do so, we build a training set from the diffraction spikes empirically identified in Megacam and DECam. We rotate the 300×300 bright-star stamps and their corresponding diffraction-spike masks using random angles uniformly chosen between $0°$ and $360°$ so that the CNN can learn to detect diffraction spikes in all directions. The CNN uses a classical semantic segmentation architecture (Ronneberger et al., 2015; Badrinarayanan et al., 2015, , see Section 5.3.1). It is shown in Fig. 5.31. It contains 8, 16, 32 and 32 feature maps built with 21×21, 11×11, 7×7 and 5×5 convolution kernels. All activation functions are ELU except on the last layer where softmax is applied to make predictions. We minimize the softmax cross entropy loss using the Adam optimizer (Kingma and Ba, 2014). In order to compensate for the unbalance between pixels with and without a spike, each pixel is weighted by $1 - p_s$ or p_s depending if it has a spike or not, where p_s is the proportion of spike pixels in the training set. We will come back to class imbalance and cost weighting issues in more details when dealing with MaxiMask.

[5]DECam images sometimes also exhibit an horizontal spike Melchior et al. (2016) of unknown origin.

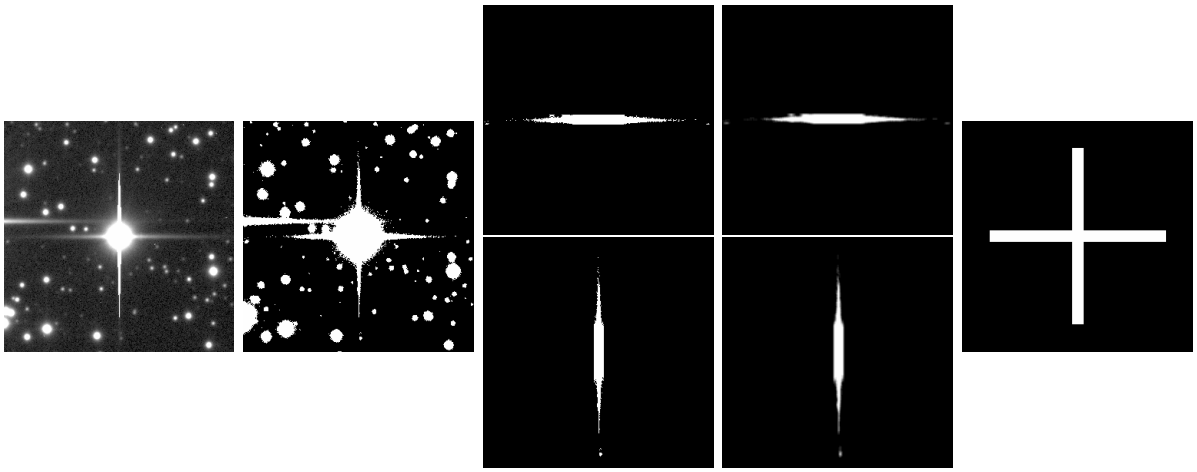


Figure 5.30: Empirical flagging process for diffraction spikes. From left to right: the source image centered on a bright star candidate, the same image thresholded, the two point-wise products, the matched-filtered point-wise products, the final mask drawn from the empirical size computed with the two previous masks.

The CNN is implemented in Python with the TensorFlow library (Abadi et al., 2016). Once trained, we run it on the HSC bright-star candidates previously detected with SExtractor to compute their diffraction-spike ground-truth mask for MaxiMask. The probabilities are thresholded based on the MC coefficient (Matthews, 1975) to create a binary mask (the MC coefficient is an accuracy estimator that takes class imbalance into account. We will come back to it in more details later when dealing with MaxiMask results). The mask is then eroded and dilated to remove any small isolated component and obtain a cleaner mask. An example is shown in Fig. 5.32.

Bright background and background

The bright-background ground-truth mask is simply obtained by thresholding the uncontaminated image at $10\sigma_U$ and dilating the resulting mask. An example of an identified bright-background pixel is shown in Fig. 5.33.

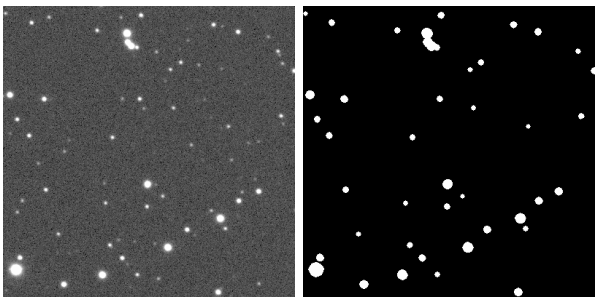


Figure 5.33: Example of bright-background pixels identified in a DECam image. Left: the input image. Right: the bright-background pixels ground-truth mask.

Finally, the background ground-truth mask is obtained by retaining the pixels that are not affected by any contaminant.

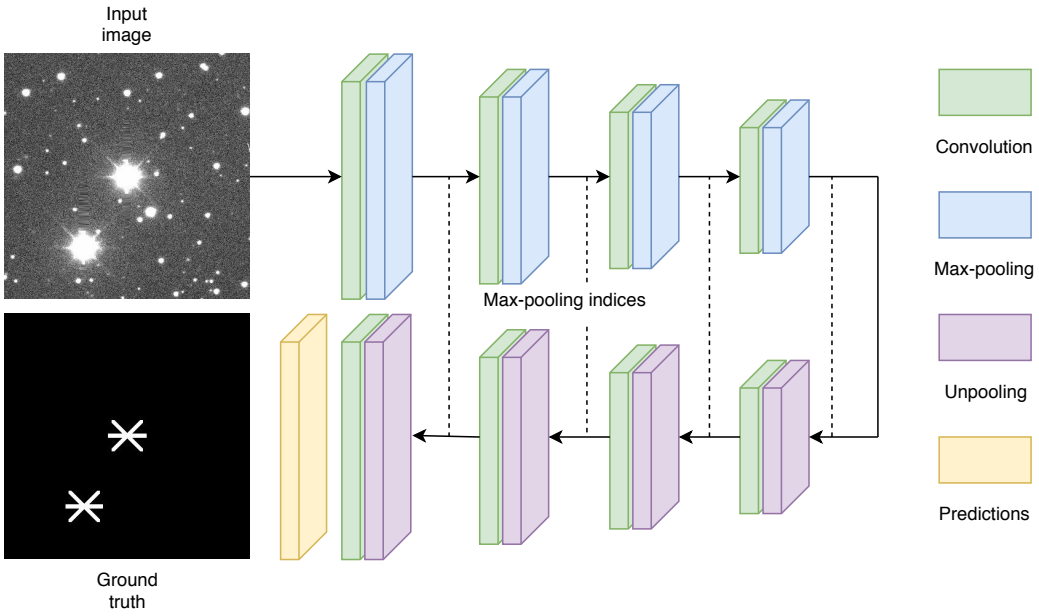


Figure 5.31: CNN used for diffraction-spike identification in HSC images. Image credit: Paillassa et al. (2020).

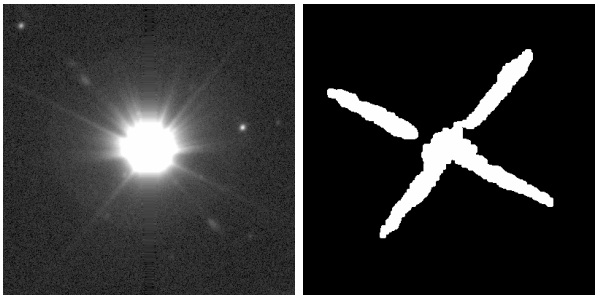


Figure 5.32: An example of cleaned diffraction-spike inference made on an HSC image by the CNN presented in Fig. 5.31. Images credit: Paillassa et al. (2020).

Dynamic compression

As mentioned in 4.2.7, normalizing the input data generally helps with training procedures. In our case, we have experimentally verified that the high-dynamic range of astronomical images is indeed an obstacle to the convergence of the neural-network. To mitigate this, we use an image normalization and dynamic-compression procedure. Our preprocessing procedure is the following:

$$\tilde{\boldsymbol{C}} = \operatorname{arsinh}\left(\frac{\boldsymbol{C} - \boldsymbol{B} + \mathcal{N}(0, \sigma_{\boldsymbol{U}}^2)}{\sigma_{\boldsymbol{U}}}\right). \tag{5.8}$$

We normalize the images by subtracting the sky background and dividing by the standard deviation of the sky-background noise. In order to make the training robust regarding small biases in the sky-background estimation, a small random offset is added between the sky-background subtraction and the division by the standard deviation.

The dynamic compression is done by applying the arsinh function, which has the interesting property of being logarithmic for extreme values and linear around zero.

Data augmentation

In order to increase the diversity of the data without gathering more images, we leverage some of the data augmentation techniques described in Section 4.2.5. We use two main data-augmentation procedures.

Firstly, we apply random rotations with angles multiple of 90° wherever it makes sense: cosmic rays, fringe patterns and nebulosity patterns.

Secondly, we rebin some images to include more critically sampled images in the training set: 50% of the uncontaminated images where the stellar FWHM remains greater than two pixels after rebinning are 2×2 rebinned. The stellar FWHMs are obtained using PSFEx (Bertin, 2011).

Examples of MaxiMask training samples

Fig. 5.34 shows some MaxiMask training samples along with their contaminant ground truths presented in the form of a single-color map. Each pixel is assigned a color depending on its class. If it belongs to several classes, it is shown in black. Pixels affected by fringes or nebulosities and another contaminant are shown with the color of the other contaminant only.

All the images contain all contaminants, except for residual fringing patterns, nebulosities, overscan regions and the inherent contaminants. Residual fringing patterns, nebulosities and overcan regions are added in 25% of the images. In addition, residual fringing patterns and nebulosities are mutually exclusive. Among the inherent contaminants, diffraction spikes are required in 75% of the images. Depending on the uncontaminated image, there are images without saturated pixels and there might be images without bright-background pixels.

5.4.3 MaxiTrack training samples

As mentioned in Section 5.4.1, we simply build the MaxiTrack training samples by gathering visually-inspected images affected by tracking error. Examples of training samples for MaxiTrack are shown in Fig. 5.35.

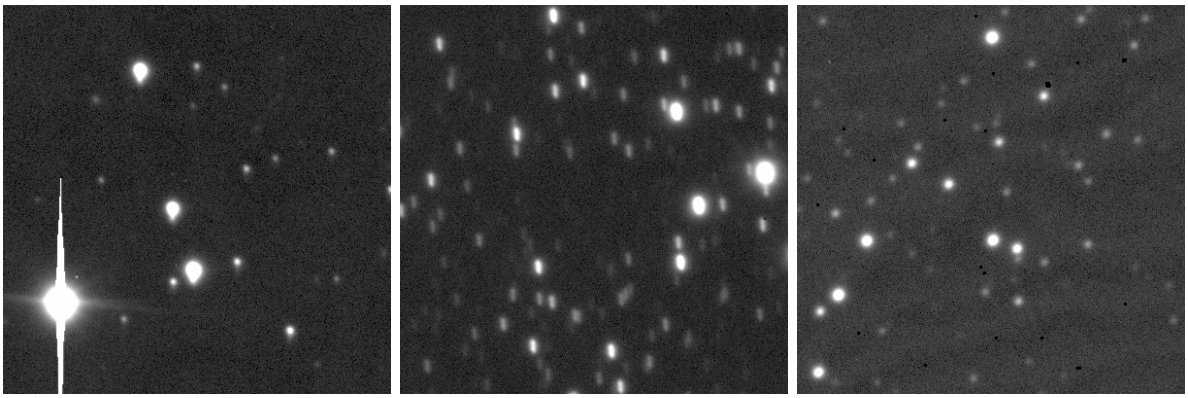


Figure 5.35: Examples of MaxiTrack training samples. Left and middle: two images affected by tracking error. Right: an image not affected by tracking error. Note that these images can contain local contaminants, like the right image that contains dead pixels.

We use for MaxiTrack the same dynamic-compression procedure as for MaxiMask (section 5.4.2).

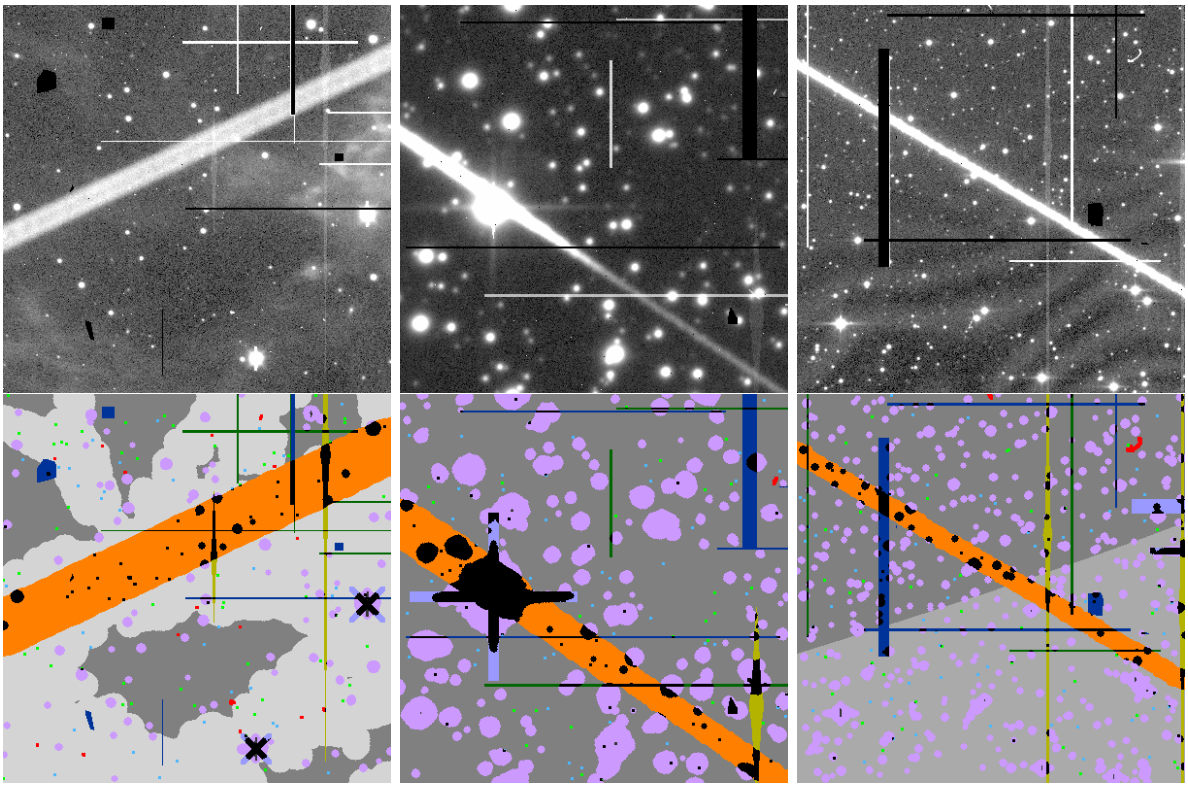


Figure 5.34: Examples of MaxiMask training samples. The contaminant ground-truth masks are all represented in a unique color map where each pixel is assigned a color depending on its class(es). The color code is: red: CR, dark green: HCL, dark blue: BCL, green: HP, blue: BP, yellow: P, orange: TRL, gray: FR, light gray: NEB, purple: SAT, lightpurple: SP, brown: OV, pink: BBG, dark gray: BG. Pixels belonging to several classes are black. Isolated hot and dead pixel masks have been dilated for visualization.

5.5 CNN architectures

5.5.1 MaxiMask CNN architecture

The MaxiMask CNN architecture is similar to the classical semantic segmentation architectures presented in Section 5.3.1 like U-net (Ronneberger et al., 2015) or SegNet (Badrinarayanan et al., 2015). It adopts the encoder-decoder architecture with a VGG backbone (Simonyan and Zisserman, 2014).

The encoder part is made of convolution and pooling layers and the decoder part is made of convolution and unpooling layers that use the max pooling indices of the same resolution feature maps from the encoder part, as explained in Fig. 5.18. All layers use ReLU activations, except the last one that uses the sigmoid to make predictions.

We also set skip connections between the two parts of the CNN: the feature maps at a given resolution in the encoder part are summed up with the unpooled feature maps of the same resolution in the decoder part. The purpose of these skip connections is to explicitly make use of the features captured in the first layers of the CNN that would not be forwarded up to the second part of the CNN.

Following Yang et al. (2018), we also use extra Unpooling Convolution Paths (UCP): the higher resolution predictions are recovered from each feature map resolution of the decoder part to form a total of five prediction maps. These five prediction maps are then fused into single-

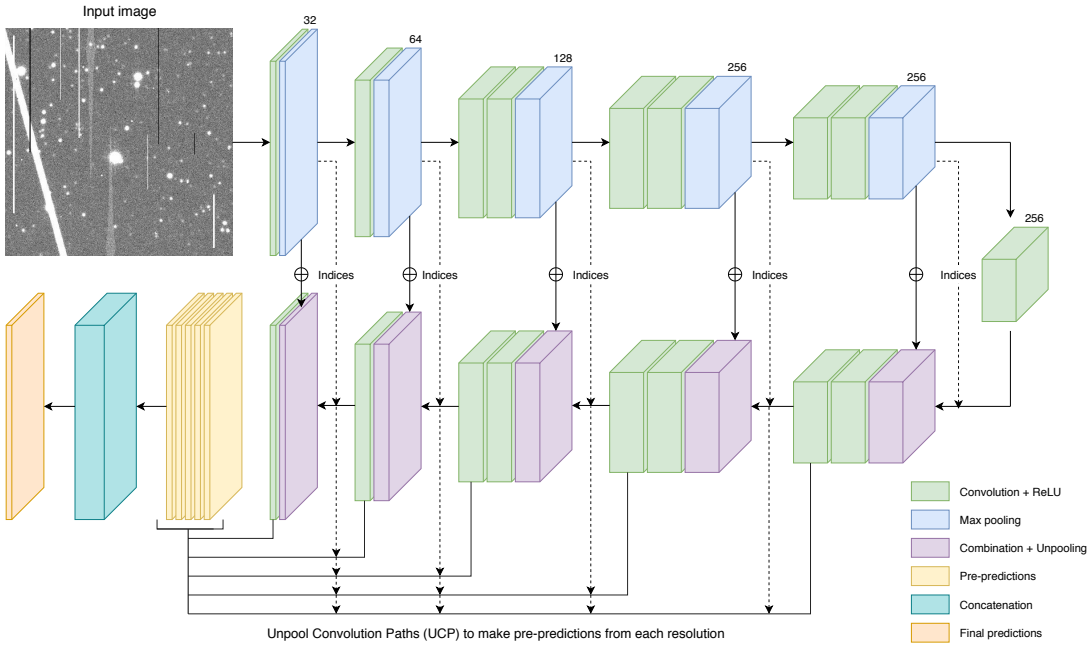


Figure 5.36: MaxiMask CNN architecture. Image credit: Paillassa et al. (2020).

prediction maps. The idea here is to extract the maximum of information from each resolution. The overall architecture is illustrated in Fig. 5.36 and Table 5.4.

5.5.2 MaxiMask loss function and class imbalance

Raw loss function

As we are using sigmoid activations in the last layer, the loss function is the common sigmoid cross entropy as defined in Section 4.2.4:

$$L_r = -\frac{1}{\text{card}(\mathcal{B})} \sum_{b\in\mathcal{B}} \sum_{p\in\mathcal{P}} \sum_{\omega_c\in\mathcal{C}} \Big(y_{b,p,\omega_c} \log \hat{y}_{b,p,\omega_c} + (1 - y_{b,p,\omega_c}) \log(1 - \hat{y}_{b,p,\omega_c}) \Big), \tag{5.9}$$

where:

- $\mathcal{B}$ is the set of batch images.
- $\mathcal{P}$ is the set of all image pixels.
- $\mathcal{C}$ is the set of all contaminant classes.
- $\hat{y}_{b,p,\omega_c}$ is the sigmoid prediction for class ω_c of pixel p of image b in the batch.
- y_{b,p,ω_c} is the ground-truth label for class ω_c of pixel p of image b, defined as:

$$y_{b,p,c} = \begin{cases} 1 & \text{if } \omega_c \in \mathcal{C}_{p,b} \\ 0 & \text{otherwise} \end{cases}, \tag{5.10}$$

  where $\mathcal{C}_{p,b} \subset \mathcal{C}$ is the set of contaminant classes labeling the pixel p of image b in the batch.

Layer	Size	UCP from each resolution			
Input	400x400x1				
Conv	400x400x32				
Maxpool	200x200x32				
Conv	200x200x64				
Maxpool	100x100x64				
Conv	100x100x128				
Conv	100x100x128				
Maxpool	50x50x128				
Conv	50x50x256				
Conv	50x50x256				
Maxpool	25x25x256				
Conv	25x25x256				
Conv	25x25x256				
Maxpool	13x13x256				
Conv	13x13x256				
Unpooling	25x25x256				
Conv	25x25x256				
Conv	25x25x256	UCP			
Unpooling	50x50x256	Idem			
Conv	50x50x256	None			
Conv	50x50x128	Idem	UCP		
Unpooling	100x100x128	Idem	Idem		
Conv	100x100x128	None	None		
Conv	100x100x64	Idem	Idem	UCP	
Unpooling	200x200x64	Idem	Idem	Idem	
Conv	200x200x32	Idem	Idem	Idem	UCP
Unpooling	400x400x32	Idem	Idem	Idem	Idem
Conv	400x400x14	Idem	Idem	Idem	Idem
Concat	400x400x70				
Conv	400x400x14				

Table 5.4: Description of the MaxiMask CNN architecture along with feature map dimensions. All convolution kernels are 3×3 and max-pooling kernels are 2×2. UCP stands for Unpooling Convolution Path.

Similar losses can be computed for each of the five prediction maps derived from the unpool convolution paths. The final total loss is a combination of the sigmoid cross entropy computed on the final prediction maps and the five sigmoid cross entropies computed from the five UCP predictions. This is done to improve the back propagation of the gradients to the lowest layers of the CNN.

There are plenty of possibilities to combine these losses. After some experiments, we adopt a combination similar to Yang et al. (2018). The best combination consist of having an equal weight between the main loss and the UCP losses and using only the smallest spatial resolution UCP losses, the latter idea being that the predictions from higher resolution provide less additional information because they are closer to the final ones. Finally, we find that adding 33% of the three smallest UCP spatial-resolution losses or 50% of the two smallest are the best tuning and I retain the first one.

Class imbalance

One of the main problem encountered with our data is the strong class imbalance between positive and negative samples. Pixels without cosmic ray hits are for example far more frequent than pixels without them, with the ratio reaching typically 1/1000 in the training set, which makes the contaminant class statistically insignificant. This can easily lead the classifier to behave as if all pixels were uncontaminated.

The class-imbalance problem has been tackled in the literature at both the data level and the algorithmic level. The main and simplest strategies used in deep learning are data sampling and cost-sensitive learning. The first one consists of resampling the data to eliminate the imbalance by either over sampling the minority class or undersampling (in the statistical senses) the majority class. For example, SMOTE (Chawla et al., 2002) synthesizes samples of the minority classes. But undersampling may imply loss of useful information and oversampling is not always feasible and may cause overfitting.

To avoid these drawbacks the second method consists of weighting the cost of the different classes according to their representation in the data set (Xu et al., 2014; Badrinarayanan et al., 2017).

More sophisticated methods have been investigated either by improving or combining these strategies. For example the max-pooling loss method (Bulo et al., 2017) combine the two previous approaches by sampling the costliest pixels and applying a weighting scheme on the remaining pixels. On the other hand, the focal-loss method (Lin et al., 2017) quickly eliminate the well classified samples so that the neural network can focus on the harder ones. The method described in (Ando and Huang, 2017) choses to oversample the data in feature space.

All these methods are quite empirical and their application can be very problem-dependent. After extensively experimenting the max-pooling loss, the focal loss and sampling methods, we conclude that cost-sensitive learning provides the best performance for our problem.

Our weighting scheme is as follows. Firstly, a weight is applied to each pixel according to its class representation in the training set, that is each pixel p of batch image b belonging to classes in $\mathcal{C}_{p,b}$ is weighted by $w_{p,b}$ defined as:

$$w_{p,b} = \sum_{\omega_c \in \mathcal{C}_{p,b}} w_{\omega_c}, \tag{5.11}$$

where each w_{ω_c} is the weight of class ω_c defined as:

$$w_{\omega_c} = \left(P(\omega_c|T) \sum_i \frac{1}{P(\omega_i|T)} \right)^{-1}, , \tag{5.12}$$

$P(\omega_c|T)$ being the fraction of pixels labeled with class ω_c in the training data set T. As some pixels may belong to several classes, the $P(\omega_c|T)$'s do not sum to one. We find that using the class proportions of the whole training set gives better performance than computing dynamically the class proportions for each batch of images.

Eq. 5.12 may not be very intuitive but this results in a weighting scheme so that the following conditions are satisfied:

$$\forall \omega_i \in \mathcal{C}, \forall \omega_j \in \mathcal{C}, \ \frac{w_{\omega_i}}{w_{\omega_j}} = \frac{P(\omega_j|T)}{P(\omega_i|T)} \ \text{ and } \sum_{\omega_c \in \mathcal{C}} w_{\omega_c} = 1, \tag{5.13}$$

that is to say that if a class is twice more represented than another one, its weight is twice smaller than the other class and the weights of all the classes sum up to one.

One problem encountered with such a weighting scheme is that non-contaminant pixels end up having a very low weight because they are statistically more represented in the training set,

leading to improper segmentation delineation. It leads to undesired effects and wrong classifications around some class of contaminants. There is for example a confusion between point-like cosmic rays (normal incidence) and hot pixels because cosmic rays are generously flagged and MaxiMask therefore tends to classify pixels around hot pixels as cosmic rays. The result is almost invisible for the loss function, as pixels around hot pixels are not contaminated and thus have a very low weight, but this behavior is problematic and not satisfactory. We therefore decided to smooth the weight maps so that non-contaminated pixels in the immediate surroundings of contaminated pixels have a higher weight. We find that smoothing with a 3×3 Gaussian kernel with unit standard deviation yields the best results.

Noting $w'_{p,b}$ the resulting weights of this smoothing, the raw loss L_r defined in Eq. 5.9 becomes:

$$L_w = -\frac{1}{\text{card}(\mathcal{B})} \sum_{b \in \mathcal{B}} \sum_{p \in \mathcal{P}} w'_{p,b} \sum_{\omega_c \in \mathcal{C}} \Big(y_{b,p,\omega_c} \log \hat{y}_{b,p,\omega_c} + (1 - y_{b,p,\omega_c}) \log(1 - \hat{y}_{b,p,\omega_c}) \Big). \tag{5.14}$$

Finally, we regularize the loss by the l2 norm introduced in Section 4.2.5, i.e., we add the l2 norm of all the N network weights to the loss L_w defined in Eq. 5.14:

$$L2_{reg} = \lambda \sum_{i}^{N} \|\boldsymbol{k}_i\|_2, \tag{5.15}$$

where the $\boldsymbol{k}_i$'s are the convolution kernel vectors and λ is a scaling factor. We find $\lambda = 1$ to provide the best results.

Other experiments and conclusions on the treatment of class imbalance Note that we also experimented the following loss function:

$$L = -\frac{1}{B} \sum_{b \leq B} \sum_{p \in \mathcal{P}} \sum_{c \leq C} \left(\frac{1}{P(\omega_c|T)} \, y_{b,p,c} \log \hat{y}_{b,p,c} + \frac{1}{1 - P(\omega_c|T)} \, (1 - y_{b,p,c}) \log(1 - \hat{y}_{b,p,c}) \right), \tag{5.16}$$

where for each class ω_c, each class and non-class pixel are weighted according to the proportion of class ω_c only. Within each class ω_c, we find that it gives slightly better results on pixels belonging to ω_c but poorer results on pixels not belonging to ω_c, i.e., we gain a bit of true-positive rate at the cost of a higher false-positive rate. Thus, we do not retain this weighting scheme.

In fact, there are two different ways to see the loss function of this multi-labeling problem. One that I call the *class* view, seeing it as the sum of each class loss L_{ω_c}:

$$L = \sum_{\omega_c \in \mathcal{C}} L_{\omega_c} = \sum_{\omega_c \in \mathcal{C}} \left(\sum_{p \in \mathcal{P}} \Big(y_{b,p,\omega_c} \log \hat{y}_{b,p,\omega_c} + (1 - y_{b,p,\omega_c}) \log(1 - \hat{y}_{b,p,\omega_c}) \Big) \right), \tag{5.17}$$

and the other one that I call the *pixel* view, seeing it as the sum of each pixel loss across all classes:

$$L = \sum_{p \in \mathcal{P}} L_p = \sum_{p \in \mathcal{P}} \left(\sum_{\omega_c \in \mathcal{C}} \Big(y_{b,p,\omega_c} \log \hat{y}_{b,p,\omega_c} + (1 - y_{b,p,\omega_c}) \log(1 - \hat{y}_{b,p,\omega_c}) \Big) \right). \tag{5.18}$$

Both result in the same loss and it just comes back to swapping the $\sum_{p \in \mathcal{P}}$ and $\sum_{\omega_c \in \mathcal{C}}$ symbols. Yet, this results in different ways to see the problem, and different ways to think about possible weighting schemes. In the *class* view, classes are seen independently one from another, which is effectively the case since we use sigmoid activations. The weighting scheme presented in Eq. 5.17

is following this view: each class ω_c is managed independently, i.e., the weights of the pixel costs for a given class ω_c only depend on the ω_c class proportion. As a result, pixels not belonging to ω_c are all weighted in the same way, by $\frac{1}{1-P(\omega_c|T)}$. However, these pixels may be very *different*, in the sense that they can belong to various other class ω'_c. Treating them equally may therefore not be ideal and this is what causes this loss to produce a solution with more false positives. On the other hand, the weighting scheme that we retain, defined by Eqs. 5.11 to 5.13 and resulting in the loss given in Eq. 5.14 is more related to the *pixel* view and accounts for all the classes of a given pixel within every class ω_c.

5.5.3 MaxiMask training

MaxiMask is trained for 30 epochs with $50,000$ images and a mini-batch size of 10. It is optimized with Adam (Kingma and Ba, 2014) and implemented using TensorFlow (Abadi et al., 2016). Fig. 5.37 shows a typical evolution of the loss function during training.

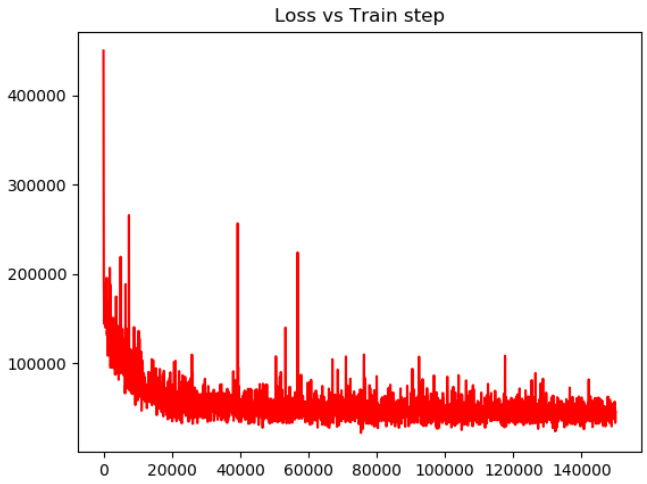


Figure 5.37: Typical evolution of the loss-function during training for MaxiMask.

I discuss some potential overfitting issues and sanity checks later in Section 5.6.1. Two examples of qualitative results after training are shown in Fig. 5.38, as well as some of their first-layer pooled feature maps.

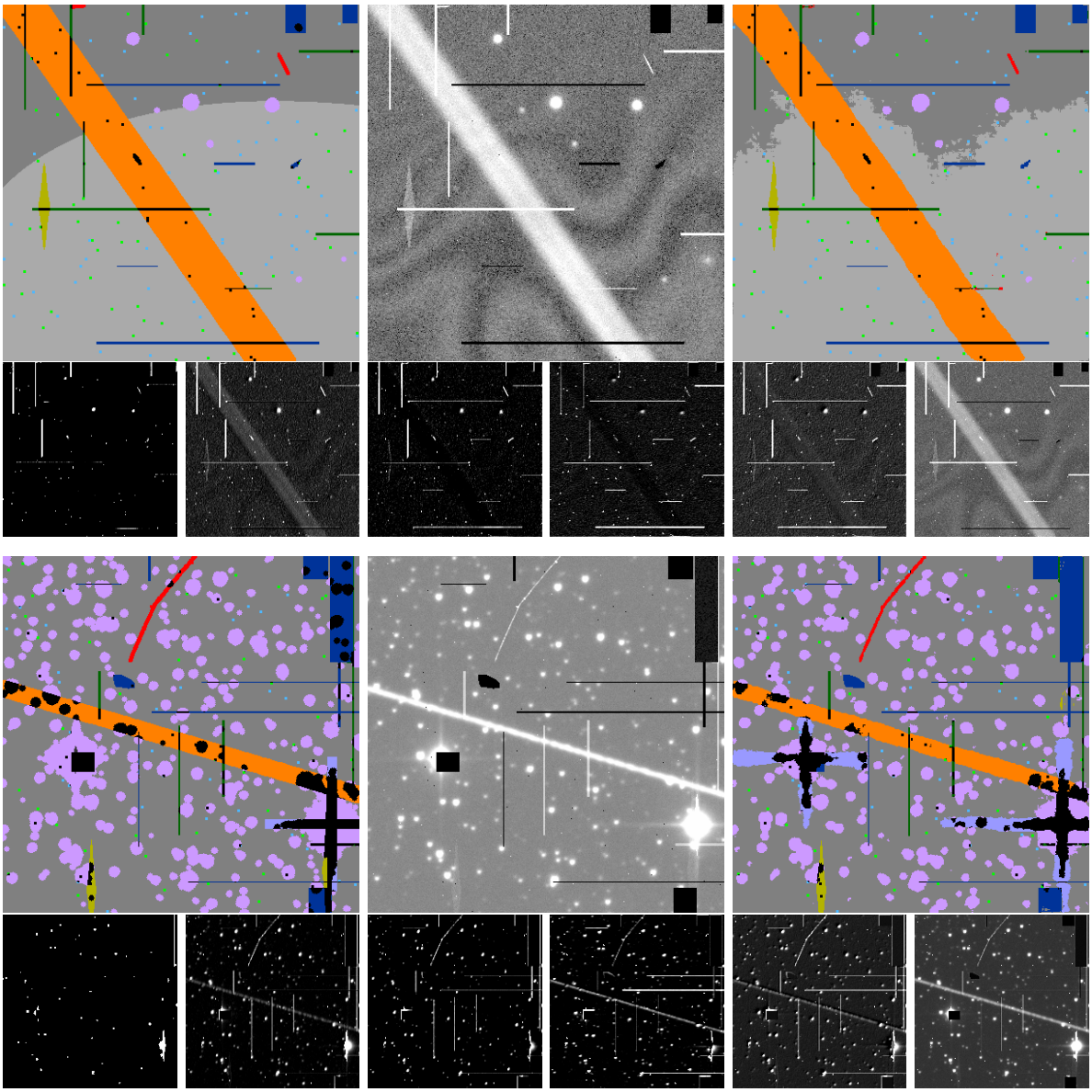


Figure 5.38: Two examples of MaxiMask results on test images. Top: ground truth, input image, predictions. Bottom: six examples of the first layer pooled feature maps.

5.5.4 MaxiTrack CNN architecture and training

The MaxiTrack CNN architecture is a classical CNN architecture for image classification (see section 4.3). It is made of several convolution and pooling layers followed by fully-connected layers (Fig. 5.39 and Table 5.5). We use dropout (section 4.2.5) to regularize the CNN in the fully-connected layers, with a dropping rate of 0.4.

The MaxiTrack loss function is the softmax cross entropy. It is optimized with Adam (Kingma and Ba, 2014) and implemented in Python using the TensorFlow library (Abadi et al., 2016). MaxiTrack is trained over 192 epochs with $50,000$ images, using a batch size of 128. Using a batch size a large as possible was found to provide better performance.

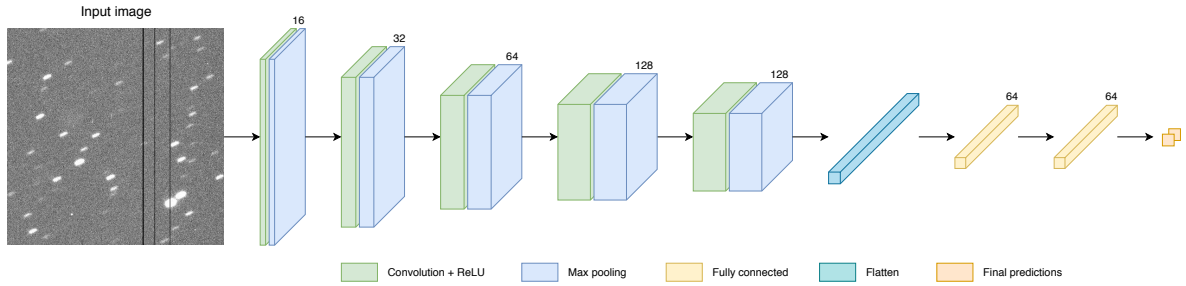


Figure 5.39: MaxiTrack CNN architecture.

Layer	Size
Input	400x400x1
Conv	400x400x16
Maxpool	200x200x16
Conv	200x200x32
Maxpool	100x100x32
Conv	100x100x64
Maxpool	50x50x64
Conv	50x50x128
Maxpool	25x25x128
Conv	25x25x128
Maxpool	13x13x128
Flatten	21632
Fully connected	64
Fully connected	64
Fully connected	2

Table 5.5: Description of the MaxiTrack CNN architecture along with feature-map dimensions. All convolution kernels are 9×9 and max-pooling kernels are 2×2.

5.5.5 Modification of the priors

As we saw in 4.2.6, in the conditions of perfect training the outputs of MaxiMask and MaxiTrack can be interpreted as posterior probabilities. Under this assumption, we can modify the priors, i.e., the expected proportion of each class, using the following:

$$P(\omega_c|\boldsymbol{x}, O) = \frac{P(\omega_c|\boldsymbol{x}, T)}{P(\omega_c|\boldsymbol{x}, T) + \frac{P(\bar{\omega_c}|O)}{P(\omega_c|O)} \frac{P(\omega_c|T)}{P(\omega_c|T)} P(\bar{\omega_c}|\boldsymbol{x}, T)} \tag{5.19}$$

$$= \frac{1}{1 + \left(\frac{1}{P(\omega_c|\boldsymbol{x},T)} - 1\right) \frac{P(\omega_c|T)}{P(\omega_c|O)} \frac{1-P(\omega_c|O)}{1-P(\omega_c|T)}}, \tag{5.20}$$

where T denotes the training set and O the observed data set we wish to run with MaxiMask or MaxiTrack. In order to modify priors, we need first to know the training priors $P(\omega_c|T)$.

The situation for MaxiTrack is particularly simple: the training set is half tracking exposures and half non-tracking exposures. Therefore, we can just apply Eq. 5.20 using $P(\omega_c|T) = 0.5$ and $P(\omega_c|O) = r$, where r is the expected ratio of exposures affected by tracking errors.

Things are not as simple for MaxiMask, because the weighting scheme that we apply in the loss function (Eq. 5.14) affects the training priors. To recover the *effective* priors $P(\omega_c|T)$, we follow Bailer-Jones et al. (2008)'s approach and use the posterior mean on the test set as an estimator:

$$\hat{P}(\omega_c|T) = \frac{1}{\text{card}(T')} \sum_{\boldsymbol{x} \in T'} P(\omega_c|\boldsymbol{x}, T'). \tag{5.21}$$

It might be counter intuitive to recover priors from the posteriors. Yet, if the posteriors are perfect, we actually get the class proportions by computing the mean of the posteriors.

5.6 Results

5.6.1 MaxiMask

We first perform a quantitative analysis of the MaxiMask classification performance on a benchmark testing set containing $5,000$ images. We also check that MaxiMask is not overfitting. Then, we check for the robustness of MaxiMask regarding the context, i.e., if it is able to perform well on images that do not contain all contaminants. Next, the cosmic-ray detection is compared to a state-of-the-art approach: LACosmic (van Dokkum, 2001; McCully and Tewes, 2019). Finally, I show some qualitative results on various instruments, including instruments not used to build the MaxiMask training set, demonstrating the ability of MaxiMask to generalize to other data.

Quantitative results

As MaxiMask acts as a binary classifier for each contaminant class, we can compute a ROC curve for each class, i.e., the true-positive rate TPR versus the false-positive rate FPR at every output threshold, as introduced in section 3.1.4:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{TP}}{\text{P}}, \tag{5.22}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} = \frac{\text{FP}}{\text{N}}. \tag{5.23}$$

The ROC curves can be found in Appendix B.1, along with the AUCs (Areas Under the Curve, see section 3.1.4). We plot the false-positive rates in logarithmic scale to better visualize the low values that we are interested in. Indeed, even though the true and false positive rates do not suffer from class imbalance (they are simply ratios), one must remember that the true-positive rate is a ratio to the positives P while the false-positive rate is a ratio to the negatives N. Thus, since N is much greater than P because of the class imbalance, it is important that the false-positive rate remains very low to ensure good performance.

For example, assuming we have $\text{P} = 1,000$ pixels in the contaminant class and $\text{N} = 159,000$ pixels in the non-contaminant class for a 400×400 pixel image, a true-positive rate $\text{TPR} = 99\%$ and a false-positive rate $\mathcal{FPR} = 1\%$ would represent 990 true positives, 10 false negatives, 157,410 true negatives, and 1590 false positives, so that there would be more false positives than true positives. This is why very low false positive rates are so important here.

However, it is important to note that performance are likely to be underestimated for the larger contaminants in the sense that it is almost impossible to recover their exact footprint down to the pixel. This is particularly true for the trails, residual fringing patterns, nebulosities, diffraction spikes and background classes.

We also draw two additional ROC curves restricted to the brighter instances of cosmic-ray and trail classes. More precisely, the ROC curves are computed by retaining only those instances that are $3\sigma_{\boldsymbol{U}}$ above the sky background. As expected, MaxiMask is able to flag the most obvious cases more easily.

In addition to the true and false positive rates, we compute two other performance metrics.

Firstly, the purity (see section 3.1.4):

$$\mathrm{PUR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}. \tag{5.24}$$

Taking back the example with $\mathrm{P} = 1,000$, $\mathrm{N} = 159,000$, $\mathrm{TPR} = 99\%$ and $\mathrm{FPR} = 1\%$, the resulting purity is $\mathrm{PUR} = 38\%$, that better highlights the poor performance of this hypothetical classifier. I draw TPR versus PUR curves for MaxiMask. These are shown in Appendix B.2.

Secondly, we compute the Matthews correlation coefficient (MCC, Matthews, 1975), as a replacement to the accuracy metric, which is not well suited to unbalanced data sets:

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}. \tag{5.25}$$

The MCC ranges from -1 to 1 (the higher, the better). It is similar to a correlation coefficient between the prediction and ground truth distributions. The MCC takes into account the four cases of TP, FP, TN, FN as well as the ratio between the amounts of positive and negative samples so that it gives a more informative and truthful score than commonly used metrics such the accuracy and the F1 score, especially with unbalanced data sets (Chicco, 2017; Chicco and Jurman, 2020). Taking back again the example with $\mathrm{P} = 1,000$, $\mathrm{N} = 159,000$, $\mathrm{TPR} = 99\%$ and $\mathrm{FPR} = 1\%$, the resulting Matthews correlation coefficient would be $\mathrm{MCC} = 0.61$, illustrating the mediocre performance of this hypothetical classifier. We compute the Matthews correlation coefficient for each MaxiMask class and plot it against the probability thresholds in Appendix B.3. The threshold giving the best Matthews correlation coefficient is annotated on each plot. Note that the probability priors are modified prior to computing these curves (unlike ROC or Purity, the MCC plots are sensitive to the priors)

Overfitting and sanity checks

The fact that the performance metrics are roughly the same on training and test data, and that we obtain very similar ROC curves and AUCs compared to testing data tells us that MaxiMask is not in the overfitting regime.

As a sanity check, we can test MaxiMask on non astronomical images (Fig. 5.40). We can observe that MaxiMask's output is consistent with the input, and that no unexpected pattern pops out. For instance, it "properly" identifies the cat's whiskers as cosmic rays and the darker areas of the image as dead pixels. The results are of course not optimal as the image does not have the same dynamic range as the astronomical image.

Figure 5.40: Example of MaxiMask inference with default priors and thresholds on a non-astronomical image.

Robustness regarding the context

In the training and testing sets, apart from some contaminants (section 5.4.2), all images contain all contaminants. Thus, we may wonder if MaxiMask is not conditioned to work only in this context. In order to verify that it is not the case, we generate a specific testing set of 1,000 images, where each image contains a single type of contaminant, and check the results. Saturated pixels and background classes are evaluated with the same testing set. The corresponding AUC for each individual testing set is reported in Table 5.6. In all the classes except residual fringing patterns and nebulosities, the AUC is slightly higher when it is measured in the contaminant specific testing sets, but we can confirm that MaxiMask is not conditioned to the specific training images context. The slight improvement may be due to the fact that there are less ambiguous cases in the contaminant specific testing sets. The corresponding ROC curves are shown in Appendix B.4.

Class	All contaminant set AUC	Single contaminant set AUC
CR	0.96927	0.98314
HCL	0.99763	0.99957
DCL	0.99872	0.99976
HP	0.99741	0.99965
DP	0.99739	0.99975
P	0.99352	0.99951
TRL	0.99511	0.99813
FR	0.98057	0.93326
NEB	0.97895	0.84575
SAT	0.99965	0.99974
SP	0.96125	0.98061
OV	0.99997	1.00000
BBG	0.98484	0.99165
BG	0.96895	0.98371

Table 5.6: AUC of each class depending on the testing set context.

Comparison with LACosmic

In order to assess the cosmic-ray detection performance of MaxiMask, we can compare it to the state-of-the-art cosmic-ray detector LACosmic (van Dokkum, 2001; McCully and Tewes, 2019).

We compare both methods using two testing sets, each with a different pixel sampling regime. The first one is made of 1,000 images that are well sampled, i.e., with a minimal FWHM of 2.5 pixels, while the second one is made of images that are critically sampled, i.e., with a maximal FWHM of 2.5 pixels. We run LACosmic with default parameters and dilate its output binary masks to match the behavior of MaxiMask on cosmic-ray flagging. The performance of LACosmic along with the MaxiMask ROC curve are plotted in Fig. 5.41. We find that MaxiMask exhibit slightly higher performance in both regimes.

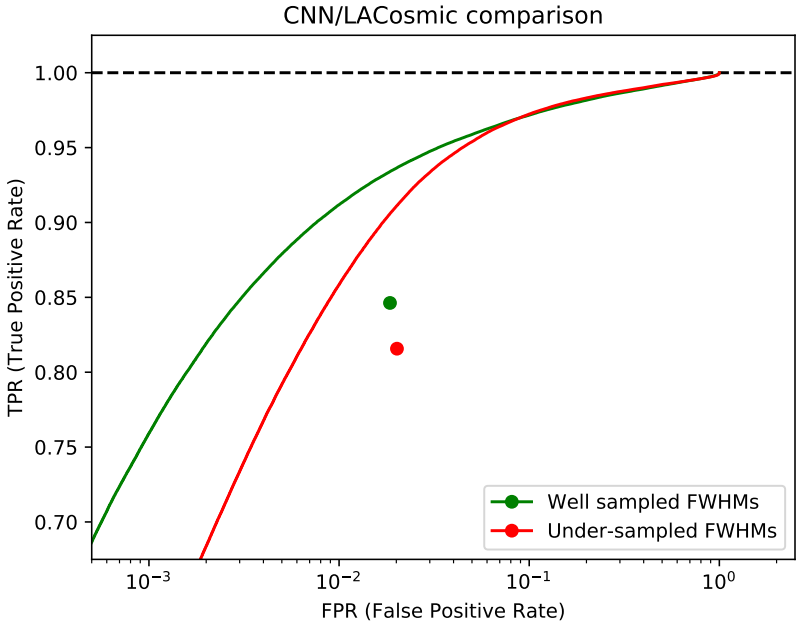


Figure 5.41: MaxiMask cosmic-ray detection performance comparison with LACosmic. Image credit: Paillassa et al. (2020).

Qualitative results on real data

The purpose of these tests is to illustrate how MaxiMask behaves on a representative set of archive and simulated image data.

DECam DECam is the archetypal wide-field, ground-based imaging instrument. Results on selected images are shown in Fig. 5.42 and Fig. 5.43, the latter showing the whole DECam mosaic with a train of satellites from the Starlink constellation. Satellite constellations are expected to increasingly affect ground-based imaging in the coming years.

HST ACS HST ACS (Advanced Camera for Surveys) exposures are rather different from the ground-based images that MaxiMask was trained on, and particularly challenging because of the undersampling and the huge amount of cosmic ray tracks. Still, as Fig. 5.44 shows, MaxiMask has no trouble making the difference between small stars and point-like cosmic rays.

Euclid VIS: Finally, we run MaxiMask on simulated Euclid SC4 images (Zoubian et al., 2014). Examples are shown in Fig. 5.45. Present Euclid simulations mainly contain cosmic rays along with ground-truth masks so that we can compute a ROC curve.

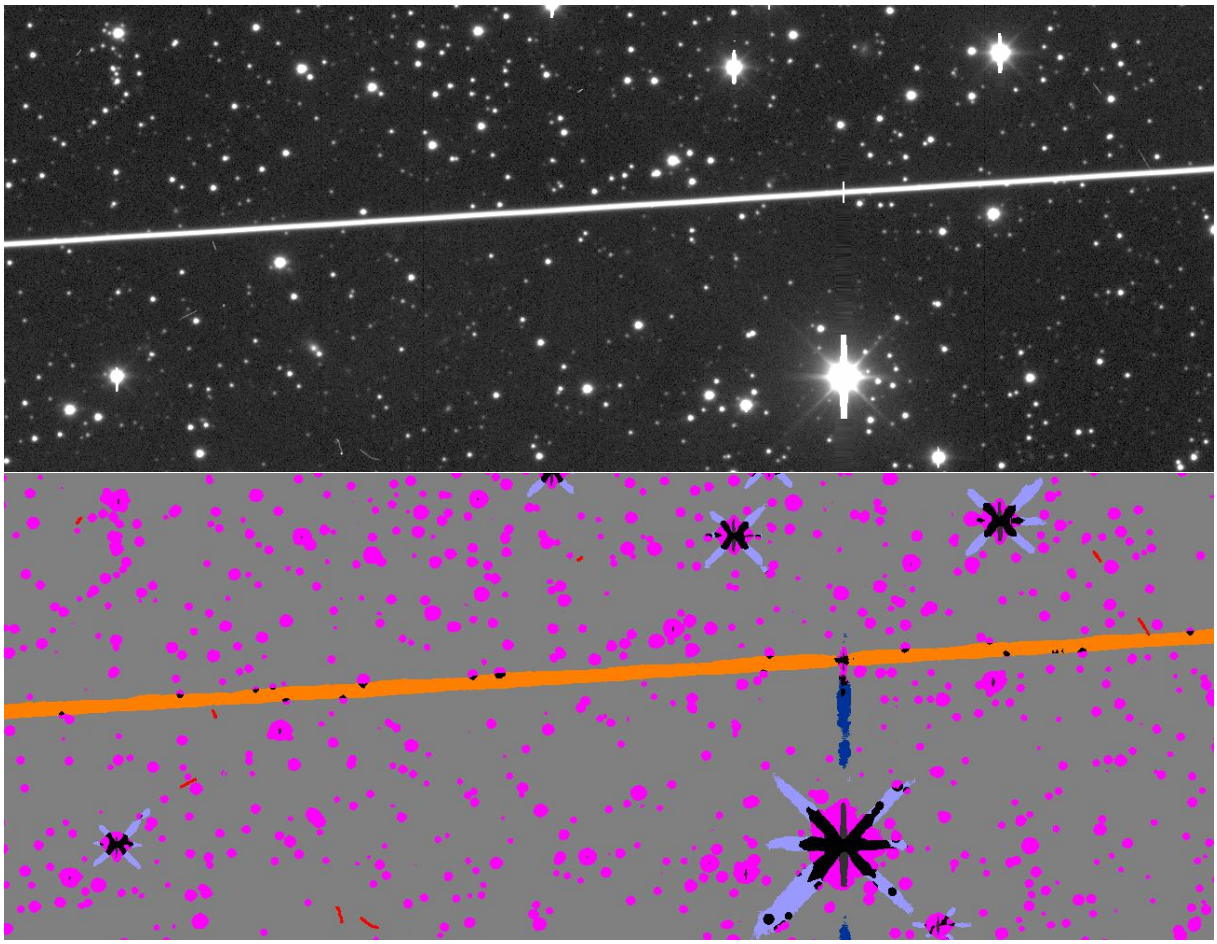


Figure 5.42: Example of MaxiMask predictions on a DECam image.

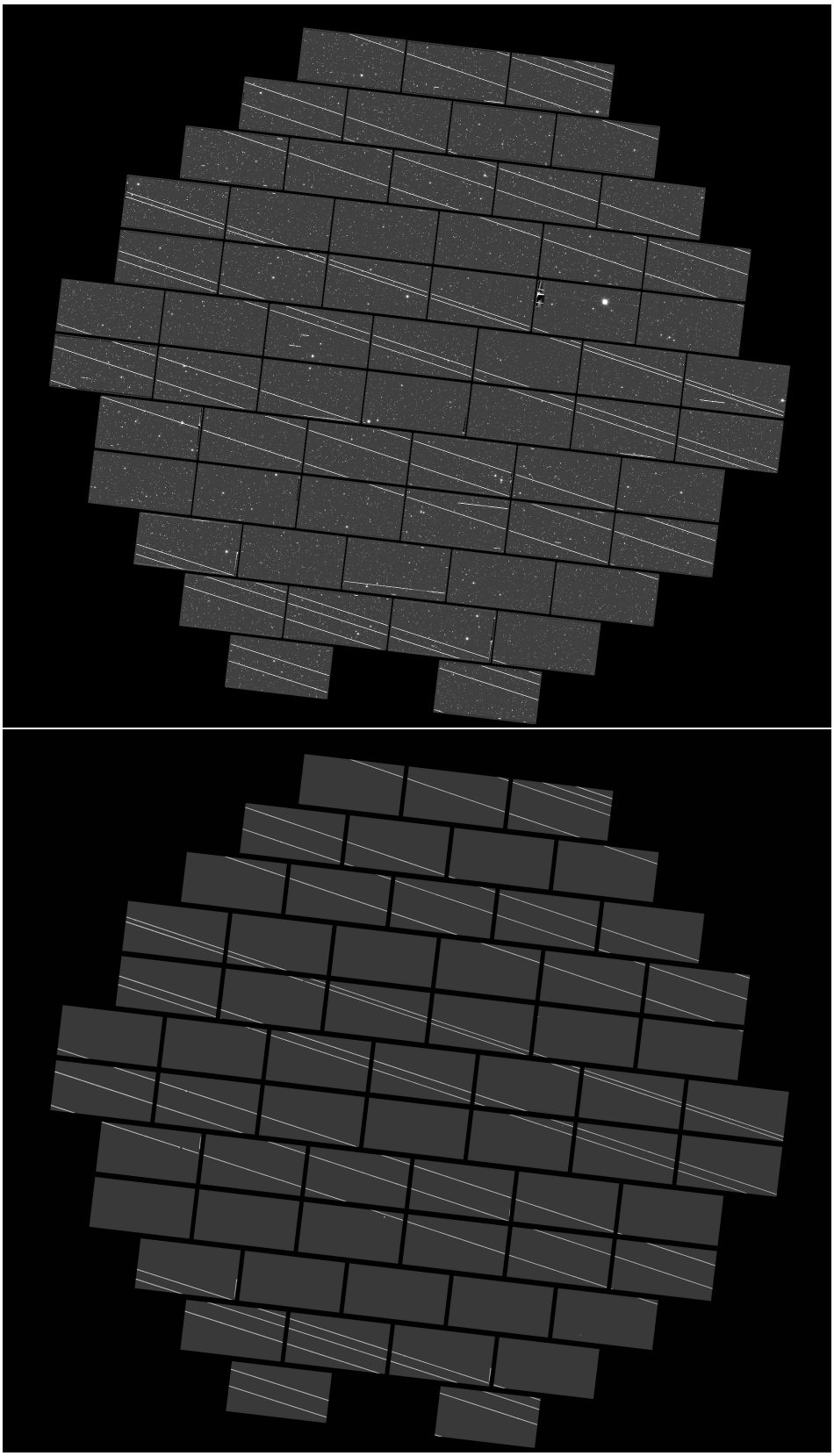


Figure 5.43: Example of MaxiMask trail prediction on the DECam mosaic contaminated by the Starlink satellites. Image credit: CTIO/AURA/DELVE (PI: Alex Drlica-Wagner).

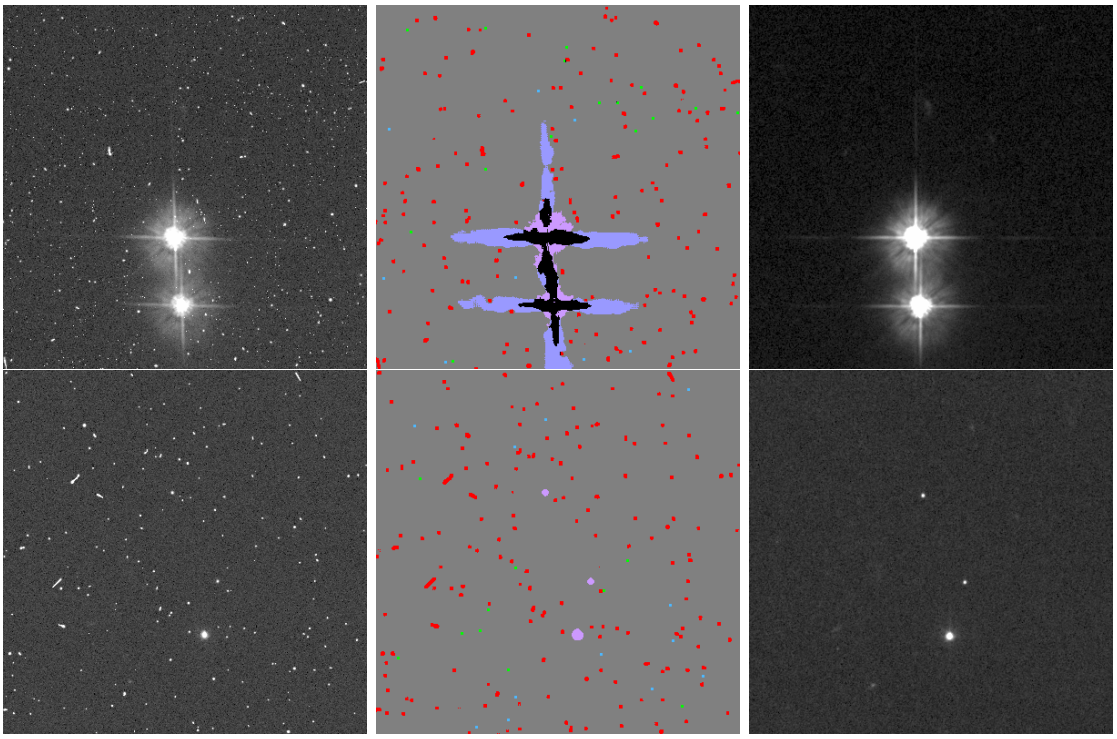


Figure 5.44: Two examples of MaxiMask predictions on HST images. From left to right: input image, MaxiMask predictions, HST pipeline processed input image.

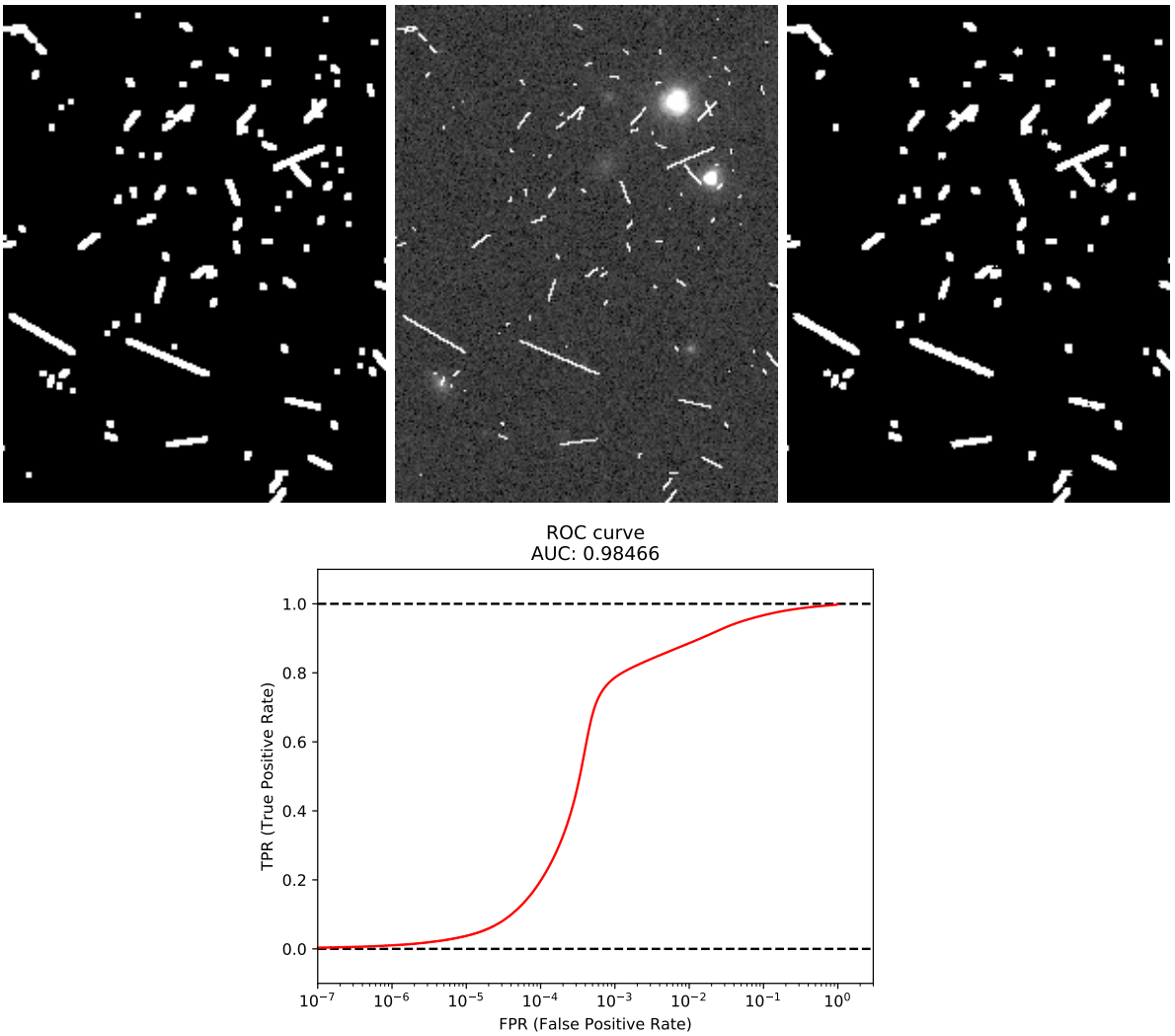


Figure 5.45: Top: qualitative example of MaxiMask cosmic-ray detection on a simulated Euclid image. From left to right: the simulated ground-truth cosmic-ray mask, the input image, the MaxiMask cosmic ray predictions. Bottom: a ROC curve computed on a whole field of 36 CCDs of 4k×4k pixels. The simulated cosmic rays are perfectly sharp. As a result, most MaxiMask false negatives correspond to perfectly point-like, vertical or horizontal cosmic ray instances.

5.6.2 MaxiTrack

Similarly to MaxiMask, we assess MaxiTrack's performance using a testing set containing 5,000 images. MaxiTrack's ROC curve for detecting tracking errors performance is shown in Fig. 5.46.

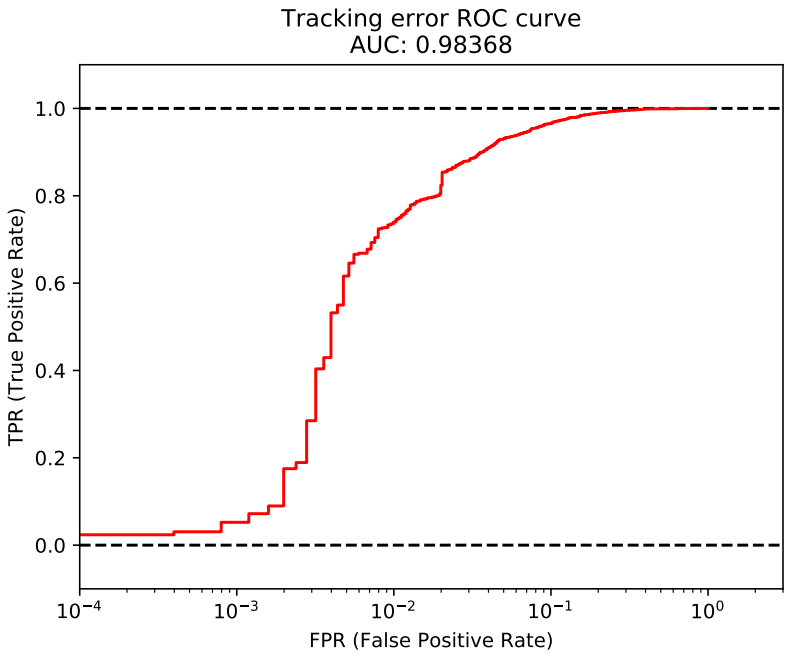


Figure 5.46: MaxiTrack ROC curve. Image credit: Paillassa et al. (2020).

5.7 The MaxiMask and MaxiTrack software packages

The MaxiMask and MaxiTrack Python packages are available at https://www.github.com/mpaillassa/MaxiMask. They were publicly released in July 2019. Both rely on the TensorFlow (Abadi et al., 2016) and Astropy (Astropy Collaboration et al., 2013; Price-Whelan et al., 2018) libraries. They can be readily applied to FITS images thanks to the provided compute graphs and weight sets.

MaxiMask outputs the contaminant probability maps from FITS images. It can manage various types of inputs such as multi-extension files and specific image extensions (HDUs). It can also process all the files within a given directory or a given file list. Unless a specific image extension is given, MaxiMask will try to process every HDU that seems to be an image, i.e., that contains 2D data. It will output a FITS file exhibiting the same HDU structure than the input file. By default, MaxiMask outputs all contaminants, updates the priors, and applies a default probability threshold to create one binary mask per contaminant. Thanks to command line arguments and simple configuration files, the user can request specific classes, change the expected priors and/or the probability thresholds. One may also disable the prior modification and/or the thresholding. Finally, an option can be set to request a single output map using a binary code for each contaminant. Such a map can easily be used as a flag map for, e.g., SExtractor. All the options and configuration parameters are stored in the output header. The code can work both with CPUs or GPUs, although the CPU version is generally much slower: MaxiMask processes about 1.2 megapixel per second with an NVidia Titan X GPU, and about 60 times less on a 2.7GHz Intel i7 dual-core CPU. Yet further optimizations could probably be done to improve the processing efficiency of both the CPU and GPU versions.

Even if MaxiMask is fully convolutional and could run on images of arbitrary sizes, it has been noticed that predictions are not exactly the same depending on the input image size. While the difference is negligible, it is not perfectly clear why it happens and it is preferable to use MaxiMask on sub-images with 400×400 pixels, which is the size of the training images. This

is why internally, the field images are divided in blocks of 400×400 pixels. Inference is made on each of these blocks, which are then stitched together again for generating the whole prediction maps. In order to avoid some boundary artifacts, we decide to space the 400×400 blocks by 200 pixels and to keep only the inner predictions in each block.

MaxiTrack can be used in the same way as MaxiMask, in the sense that it accepts multi-extension files, specific image extensions, a directory or a file list as input. Yet, it does not output prediction maps but simply a text file that contains the probability that the input image(s) are affected by tracking errors. Priors can also be modified. MaxiTrack also runs inferences on blocks of 400×400 pixels internally. The resulting tracking-error probability for a whole field is then the mean of all the block predictions. This increases the robustness of MaxiTrack as few errors on some image blocks may not impact dramatically the final result. MaxiTrack runs at 60 megapixels/s with an NVidia Titan X GPU and is 9 times slower on a 2.7GHz Intel i7 dual-core CPU. In order to use it even faster, the user can specify a smaller number of HDUs to compute the tracking error probability. In this case, MaxiTrack will randomly pick the requested number of HDUs to make its prediction.

5.8 Conclusion and perspectives

We have developed MaxiMask and MaxiTrack, two fully convolutional neural networks that automatically identify a wide range of contaminants by scanning astronomical images. MaxiMask and MaxiTrack were trained using a custom-built data set from a large collection of ground-based images. To train MaxiMask, we developed a solution to mitigate the problem of class imbalance in the context of semantic segmentation. Thanks to the probabilistic outputs of MaxiMask and MaxiTrack, one can set appropriate priors and thresholds to target specific true and false positive rates depending on scientific requirements. To our knowledge this is the first time that such a generic approach is taken to tackle this problem. We have shown that MaxiMask and Maxitrack perform as well and even better than previous state-of-the-art defect detectors, while requiring no or little tuning. Both codes are already running in production on exposures from the COSMIC-DANCE survey (tens of thousands of exposures so far) and MaxiMask is currently being tested by the Dark Energy Survey data management team for the identification of satellite trails.

A valuable extension to MaxiMask would be an inpainting module for correcting the pixels affected by contaminants, instead of just flagging them.

However, MaxiMask still misses several important contaminants. In particular, we plan to include bright star halos, optical ghosts, as well as reflections and scattered lights in a future version. This type of contaminant is particularly harmful, especially in wide-field cameras such as HSC (Fig. 5.10), but preparing ground truths for such contaminants will be challenging. Other defects not yet flagged by MaxiMask include crosstalks (Figs. 5.4 and 5.5), and saturation patterns in infrared cameras (Fig. 5.2). Crosstalks create transient patterns in images, which may interfere with searches for astrophysical transients such as supernovae or counterparts to gravitational wave events. They also affect high precision astrometric and photometric measurements. These contaminants should in principle be easily added to MaxiMask provided that a training set is available.

MaxiMask seems to perform well right out of the box on exposures from spaced-based instruments, although performance is probably degraded due to data mismatch. Use for production with, e.g., Euclid data would require re-training with image simulations containing defects specific to the mission. However, this should not require a huge amount of work, as ground truths will readily be available.

Finally, as far as our project is concerned, the main benefits from this work are not Maxi-

Mask and MaxiTrack themselves, but the training set, the neural network architectures, and the training procedures that we will partly reuse in the next chapter.

Chapter 6

Source detection

In Chapter 3, we saw that the matched filter is the optimal linear filter for detecting isolated sources of known profiles in the background noise limited regime. Currently, it is the most widely used method for source detection in image analysis pipelines. Unfortunately, as we also saw in 3, the matched filter approach becomes inefficient for sources with varying shapes or scales, or when images are contaminated or crowded.

After having addressed the question of contaminants in Chapter 5, I will now focus on the issue of source deblending and detection. Compared to MaxiMask and other generic semantic segmentation methods which do not distinguish between individual objects of the same class, we now have the additional requirement that the source detector must be "instance-aware", i.e., it must be able to detect and isolate each source individually.

However, it must also be able to deblend and recover overlapping sources, which makes the design of the source detector more complicated than that of conventional instance-aware segmentation CNNs (a similar problem occasionally dealt with in natural scenes is object occlusion).

In this chapter, I first explore several approaches, starting with a simple identification of centroids. I review existing techniques based on instance-aware semantic segmentation and examine their relevance in the context of the detection of astronomical sources. I then present our solution for multiscale object detection and deblending, and outline the architecture of a prototype CNN in details. After describing the construction of the training data set, I assess the performance of the prototype in different observation regimes, comparing the results with those obtained from a conventional source extraction package. Finally, I conclude this chapter by pointing out some limitations of the current prototype and discussing future developments.

6.1 Detecting source centroids

This project finds its origin in a 2016 internship (Paillassa and Bertin, 2019). At the time, we circumvented the instance-aware requirement by identifying sources by their centroids, via semantic segmentation. Indeed, each source can be individually identified by its centroid and any source blend can be disentangled as long as the blended source centroids remain distincts. Despite the simplistic CNN architecture and training data sets that were just simulations of stellar fields, we showed that a unique CNN could outperform SExtractor on all test images.

Taking back this approach with a more complex CNN and training set, we found that we could achieve very good detection performance for point-like sources even in crowded fields. In particular, applying loss sampling to address the strong imbalance between centroid and non-centroid pixels was found to work well. However, we could not find any suitable configuration to produce satisfactory results with extended sources. This is because the centroid is not the best feature for characterizing extended sources. More generally, a single point does not characterize

well extended sources since they do not represent a particular intensity peak, nor any particular feature down to the pixel level.

In addition, using source centroids poses other problems. Since the source centroid is defined by the barycenter of the source pixels, it may not fall perfectly on a given pixel and a rounding must be done to assign it to a given pixel. This strongly non-linear behavior can certainly be learnt by a CNN but it is not using its capacity to good effect. Furthermore, a small error in the position of the centroid prediction would have a big impact on the CNN cost function, while a precision down to the pixel is not required for detecting an extended source. These issues quickly had us move away from this approach and turn to more classical instance-aware object detection techniques.

6.2 Deep learning methods for instance aware object detection

With the advancement of deep learning, instance-aware object detection and segmentation has become a very active and profuse area of research. In the following, I present the most iconic approaches for instance-aware object detection and discuss their potential application to source detection. I divide the approaches in three categories: two-stage detectors, one-stage detectors and other approaches.

6.2.1 Two-stage detectors

Many approaches for instance-aware object detection are based on two-stage detectors. One of the most representative two-stage detector is Faster R-CNN (Ren et al., 2015), which is an improvement of Fast R-CNN (Girshick, 2015) and R-CNN (Girshick et al., 2014). It works in two steps. The first step uses a CNN known as Region Proposal Network (RPN), which produces object region proposals in the form of bounding boxes. The positions of the bounding boxes are predicted by a fully convolution neural network, via regression. In a second step the proposals are classified by another CNN that shares feature maps with the RPN. Before being fetched into the classification CNN, bounding-box proposals are resized to a fixed size via the Region of Interest pooling layer (RoI pooling). Note that a proposal may be rejected in the second step. An illustration of the Faster R-CNN architecture is shown in Fig. 6.1.

Faster R-CNN has also been extended to Mask R-CNN (He et al., 2017) to perform instance-aware object segmentation by simply adding a segmentation CNN in parallel with the classification CNN of the second stage.

These types of approaches build upon previous work in the literature concerning object detection via bounding boxes. However, bounding boxes are not very suitable to characterize astronomical objects, as the latter do not have clear boundaries or can be made of multiple components. In addition, the RPN introduces several limitations. First, it tends to produce many close proposals for the same object. In order to eliminate spurious proposals, the non-maximum suppression procedure is applied (Canny, 1986). Based on a matching criterion with predefined bounding boxes called anchors, it consists in ignoring the majority of proposals to retain only one per object. This procedure is not suitable for deblending because it cannot really distinguish if two close proposals relate to a single object or to different objects. Second, the RPN is constrained to search for object bounding boxes that correpond to predefined bounding boxes (the anchors). Even if one can predefine anchors of different shapes and scales, it is still a limitation as their number needs to be limited. In addition to that, the RoI pooling layer can introduce quantization of the feature maps and limit segmentation capabilities. Even though the operation has been improved as RoI Align in Mask R-CNN (He et al., 2017), bilinear interpolation is used instead which is far from optimal for achieving precise object segmentation. Therefore, these techniques do not seem very suitable for our source detection problem.

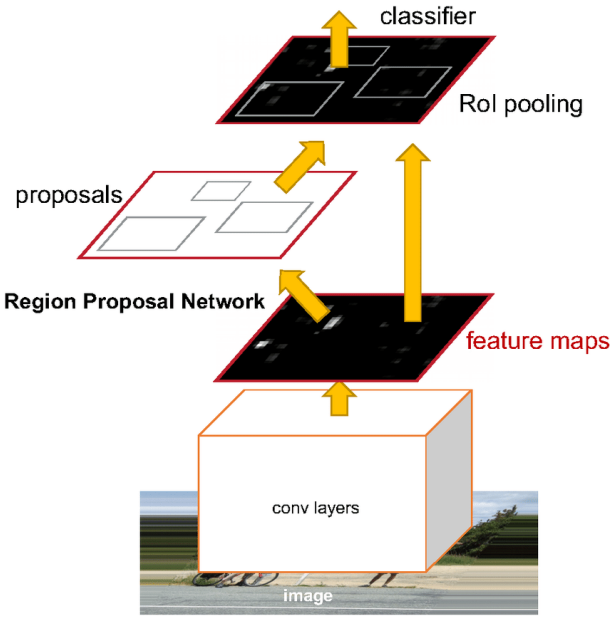


Figure 6.1: Faster R-CNN (Ren et al., 2015) architecture. Image credit: Ren et al. (2015).

6.2.2 One-stage detectors

The main other instance-aware object detection methods are one-stage detectors, such as YOLO (You Only Look Once, Redmon et al., 2016). The principle of YOLO is to avoid the two-step approaches from section 6.2.1 by detecting and classifying objects simultaneously. To do so, it divides the input image into blocks. In each block, bounding-box proposals are predicted in the same way as an RPN. At the same time, each block is also assigned an object class. This makes it possible to match the classes with the bounding boxes by their location. An illustration of YOLO is shown in Fig. 6.2.

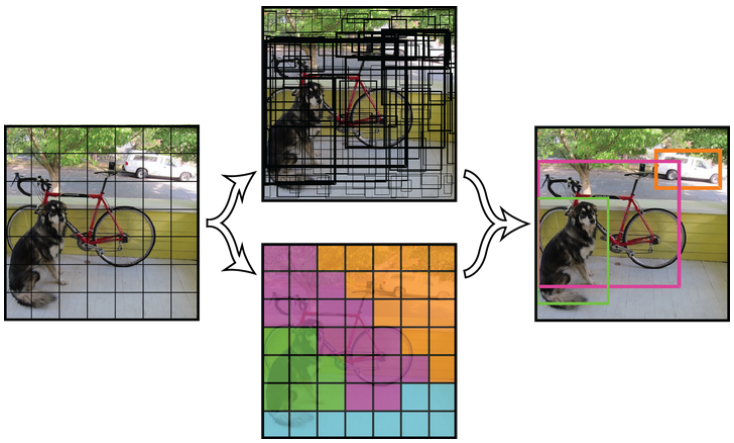


Figure 6.2: YOLO (Redmon et al., 2016) architecture. Image credit: Redmon et al. (2016).

Note that YOLO has been improved as YOLO9000, a.k.a YOLOv2 (Redmon and Farhadi, 2017), YOLOv3 (Redmon and Farhadi, 2018) and even more recently YOLOv4 (Bochkovskiy et al., 2020). Since YOLO9000, it also uses box anchors.

The advantages of YOLO are that it is fast and it can use the image context in a better way than Faster R-CNN because all the object-prediction procedure is obtained from the whole image. However, it can only predict a limited number of objects in each block and it is bad

at detecting small objects, which are two important drawbacks in the context of point-source detection.

6.2.3 Other approaches

Besides the two main instance-aware object detection techniques above, alternative methods exist, in particular methods that do not use bounding boxes.

Recurrent instance segmentation

A potentially interesting instance-aware object detection method is to use recurrent neural networks to segment objects one by one (Romera-Paredes and Torr, 2016). Recurrent neurons exhibit two properties that are used in recurrent instance segmentation. First, they can handle sequential inputs and outputs. Second, they use a memory state to store features encountered in the input sequence. In the recurrent instance segmentation method, feature maps are extracted from the input image using a CNN and are fed to recurrent neurons. The recurrent neurons then predict one object at a time as well as a score indicating whether there are still objects left in the image. These can loop an arbitrary number of times to sequentially detect all objects in the image.

However, this behavior imply some complications in the loss function. Indeed, as objects are predicted in an arbitrary order, it is necessary to know to which ground truth object corresponds a given predicted object. When having N predicted objects and M ground truth objects, the question to be answered is: given a measure of cost between a pair of predicted and ground-truth objects, how to match the predicted and ground-truth objects so that the global cost is minimized ? This problem is an assignment problem, that can be solved via the Hungarian algorithm (Kuhn, 1955). It is used in the recurrent instance segmentation approach.

Even if this approach is original, it is quite complex and it can be very slow because the processing time depends directly on the number of objects in the image.

Objects as points

Another possible approach to avoid bounding boxes consists in modeling objects as points (Zhou et al., 2019). An object is defined by its center and other properties such as the size or orientation. The latter are regressed from the features near the detected center. Object centers are predicted via keypoint estimation, a widely used technique in, e.g., human pose estimation techniques. The ground truth to train the CNN consists of a heatmap made of Gaussians at the center of each location. During inference, peaks in the heatmap are considered as object centers. Regression maps of the same size than the keypoint heatmap are used to predict the object properties.

One limitation of a direct application of this approach to source detection is that objects are detected in feature maps at a lower spatial resolution than the initial input image spatial resolution. This greatly limits the number of objects that can be detected and requires additional offset regression to recover the exact position of the detected objects in the initial spatial resolution. Also, regressing the object properties in a map at the same resolution as the heatmaps does not seem optimal as only few values corresponding to the detected peaks are of interest. Finally, the ground truth keypoints can overlap so there may be ambiguities to separate objects. Detecting peaks in the heatmap is equivalent to non maximum suppression and must to be finely tuned.

Deep coloring

Deep coloring (Kulikov et al., 2018) may be the most promising method to segment and deblend sources directly. In this approach, objects are segmented in different colors, i.e., in different

output maps or classes. By using a color assignment criterion based on the proximity of objects, close objects are constrained to be identified in different colors.

More precisely, let's assume we have N objects in an image, each having its segmentation mask $M^{(k)}$. Another mask $M^{(k)}_{halo}$ is defined as:

$$M^{(k)}_{halo} = \text{dilation}(M^{(k)}) - M^{(k)}. \tag{6.1}$$

$M^{(k)}_{halo}$ represents a more or less the close neighborhood of object k, that is to say pixels that, if they belong to another object, should be identified in another color.

Assuming that there are C colors, the CNN predicts $C+1$ maps: C colors where objects are flagged and one for the background. Let's call y the prediction, $y(c,p)$ designing the prediction of pixel p of color c. The background prediction is the last map: $C+1$.

During training, in order to compute the loss of each object k, the color c_k is found using the following color criterion:

$$c_k = \arg\max_{c}^{C} \frac{1}{|M^{(k)}|} \sum_{p \in M^{(k)}} \log(y(c,p)) + \mu \frac{1}{|M^{(k)}_{halo}|} \sum_{p \in M^{(k)}_{halo}} \log(1 - y(c,p)). \tag{6.2}$$

In other words, the criterion favors the coloring of neighboring objects in a different color.

Then, the cost function of the network is simply:

$$L(\boldsymbol{x}, \theta) = -\sum_{k}^{N} \frac{1}{|M^{(k)}|} \sum_{p \in M^{(k)}} \log(y(c_k,p)) - \sum_{p \in back} \log(y(C+1,p)), \tag{6.3}$$

which is the softmax cross entropy where the cost of each object is dynamically computed over its color map defined by the criterion of Eq. 6.2.

The main obstacle to adapting this approach to source detection is that the softmax activation function requires that pixels be assigned a single object. Therefore, in the present state it is impossible to manage blends where pixels must belong to several objects simultaneously.

To circumvent this, one could use a sigmoid activation function so that a given pixel could belong to several objects and different colors. However, sigmoid activations pose a problem because all colors become independent and it is possible for the network to flag all the objects in all colors.

We are looking into an adaptation of the loss function to sigmoid activations by adding a term requiring that when an object is identified in a given color, it should not be identified in any other colors (except for the pixels that belong several objects). Yet, we haven't found any configuration or weighting between the different cost function terms leading to satisfactory results. It seems that when the dependency of each color is broken, the CNN cannot converge to a good solution. It is possible that further experiments will make this method work for source detection but for the time being we will take another direction.

6.3 Deep learning applications to the detection of astronomical sources

Despite the fact that we have not yet found any suitable approach directly applicable to source detection, there have been some deep learning experiments in the field, whether or not following the aforementioned approaches.

Some of the techniques described above have been directly applied to source detection. For instance, González et al. (2018) detect and classify galaxies with YOLO (Redmon et al., 2016),

Burke et al. (2019) apply Mask R-CNN (He et al., 2017) to source detection and segmentation, and Jia et al. (2020) use Faster R-CNN (Ren et al., 2015) to detect point-like and streak-like sources. However, in addition to having the shortcomings already mentioned, these solutions were all designed to work with specific data. Moreover, González et al. (2018) are limited to galaxies while Jia et al. (2020) are limited to point-like and streak-like sources. Therefore, none of these applications meet our requirement to design a universal detector.

Another method (Hausen and Robertson, 2020) uses semantic segmentation to classify pixels depending on the morphological type, such as spheroid, disk, irregular, point-like source and background. Yet, the background segmentation map is post-processed with watershed transforms to segment and deblend objects, so that the approach falls back to the empirical methods described in Chapter 3. Other types of approaches solely use semantic segmentation to identify sources by their centroid, similarly to our experiments mentioned earlier (Paillassa and Bertin, 2019). They are effective at detecting point-sources and have been particularly used with radio images (Vafaei Sadr et al., 2019; Lukic et al., 2019).

Deep learning techniques have also been leveraged to specifically tackle deblending. However, as already discussed in Section 3.1.5, there are two levels of deblending: the detection level (detecting sources in crowded regions) and the measurement level (correcting measurements for the presence of close neighbors). Although we are interested in deblending at the detection level, all the existing deep learning approaches so far have been dealing with deblending at the measurement level.

For example, Lanusse et al. (2019) deblend with autoregressive models (Oord et al., 2016; Salimans et al., 2017; Chen et al., 2018). The method is designed to make use of prior knowledge, which does not match our requirements in this work. Using more classical semantic segmentation methods, Boucaud et al. (2020) segment and measure the photometry of pairs of blended galaxies. Even more recently, Arcelin et al. (2020) use variational auto-encoders (Kingma and Welling, 2013) to recover the individual images of the brightest source from a blended image.

To conclude this overview of existing instance-aware detection techniques, it seems that none at this stage fits perfectly our requirements, namely to be versatile and robust to contaminants, while being able to deblend close or overlapping sources at the same time.

6.4 Our solution

6.4.1 A multiscale approach

The basic principle of our approach to detection/deblending is to identify each source by a single component footprint that is small enough to be separated from other close source footprints by connected component labeling (Rosenfeld and Pfaltz, 1966).

In complement to this procedure for isolating sources, we set up a multiscale detector: sources of different scales are identified in different output maps. While this complicates the building of training samples because sources are assigned different scales, this provides a natural deblending scheme for overlapping objects with dissimilar sizes. For instance, a point-source located in the wings of an extended source can be naturally deblended as both are identified in different output maps.

The core of our algorithm resides in the definition of the source footprints and in the source scale affectation procedure.

6.4.2 Source footprints

A source footprint can be defined in different ways. The main difficulty is in defining footprints that remain meaningful enough to the CNN so that even the most peculiar sources such as

asymmetric multi-component or non-convex sources can be detected.

We identify two relevant and easy ways to define footprints:

- Retaining the brightest pixels forming some percentage p_F of the total flux F of the source.
- Retaining the pixels brighter than some fraction f_{I_M} of the maximum pixel intensity I_M of the source.

In both cases the footprint is guaranteed to overlap bright regions of the object. This would not be the case if, e.g., the footprint was defined by pixels around the source centroid, or by a rescaling of a larger footprint, because of a possibly non-convex object shape.

Footprint area (defined by the number of footprint pixels) is critical, and can be adjusted using p_F and f_{I_M}. Smaller footprints improve the deblending capabilities of the CNN. However they make the recovery of the most diffuse sources harder, and they increase the relative cost of footprint errors on the CNN loss function. Conversely, for larger footprints the CNN is less affected by small positional errors but deblending capabilities are reduced.

We also note that favoring too much deblending with really small footprints may induce a strong prior on the source shapes, which is not ideal in the optic of designing a universal detector. For instance, two close circular extended sources can appear as a single source with an elliptical shape in an image. In this case, it can be difficult to decide if there are two close circular sources or a single elliptical source. We thus may wonder if the detector should predict it as a single source or two sources and we can intentionally favor one behavior by tuning the sizes of the ground truth footprints.

If the source footprints overlap, we may use smaller footprints so that they become distinct by connected component labeling, the risk being that the detector may overdeblend other elliptical looking sources that should not be. After experimenting with various footprint areas, we find that footprints defined by $p_F = 25\%$ represent a good trade-off. We also find that dilating the footprints of the two first scales give better performance as it mitigates imbalance issues.

6.4.3 Source scale assignment

Within our multiscale framework, each source must be assigned a particular scale. We have experimented with different assignment procedures based on the area of the source footprints. One could think of sorting all sources by footprint area and making one group for each of the three scales, each containing an identical number of sources. However this relies on the distribution of source sizes, which in practice is extremely field- and instrument-dependent. We eventually opted for an arbitrary criterion based on dyadic scales in footprint size (prior to dilation). We first define a maximum footprint area of 9 pixels for the first scale: sources which footprint contain less than 9 pixels are assigned to that scale. The following footprint limits are obtained by multiplying the maximum area of the previous scale by 4. Sources with footprints containing less than 36 pixels are thus assigned to the second scale while the third scale includes sources with footprints having more than 36 pixels and so on. Example of footprint ground truths are shown in Fig. 6.8.

6.4.4 CNN architecture

The CNN detector architecture is a classical semantic segmentation architecture (Ronneberger et al., 2015; Badrinarayanan et al., 2017), as described in Section 5.3.1. The first part of the CNN is made of convolutional and pooling layers. The second part of the CNN is made of unpooling and convolutional layers to recover the initial spatial resolution and make pixel-wise predictions. We follow the unpooling procedure which consists of using the max-pooling indices of the first

part of the network to upsample the feature maps (Badrinarayanan et al., 2017). We also use skip connections between the two parts of the CNN: we add the feature maps of the first part of the CNN at the same spatial resolution in the second part of the CNN. The hyperparameters of the network are based on the VGG architecture (Simonyan and Zisserman, 2014). The detailed architecture of the CNN is shown in Fig. 6.3 and described in Table 6.1.

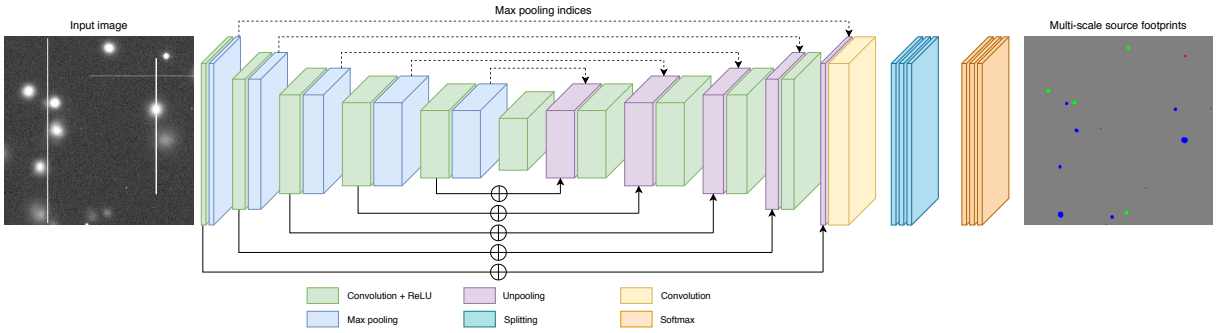


Figure 6.3: CNN architecture of the multiscale source detector.

Layer	Size
Input	400x400x1
Conv	400x400x32
Maxpool	200x200x32
Conv	200x200x64
Maxpool	100x100x64
Conv	100x100x128
Maxpool	50x50x128
Conv	50x50x128
Maxpool	25x25x128
Conv	25x25x128
Maxpool	13x13x128
Conv	13x13x128
Unpool	25x25x128
Conv	25x25x128
Unpool	50x50x128
Conv	50x50x128
Unpool	100x100x128
Conv	100x100x64
Unpool	200x200x64
Conv	200x200x32
Unpool	400x400x32
Conv	400x400x6

Table 6.1: Description of the CNN source detector architecture along with feature-map dimensions. All convolution kernels are 3×3 and max-pooling kernels are 2×2.

6.5 Image simulations

Our training and testing data sets consist of images simulated from scratch using noise-free images of isolated sources. The absence of noise makes it possible to generate the ground truth of an arbitrary footprint for every individual source. Noise and contaminants are added later to generate realistic astronomical images. In summary, our image simulation process contains the following steps:

- Adding the noise-free images of isolated sources in a single image.
- Simulating and adding a sky background flux.
- Making the Poisson noise realization.
- Applying a gain transformation.
- Adding Gaussian readout noise.
- Adding possible contaminants.

Fig. 6.4 illustrates the image simulation pipeline. Simulated image parameters are listed in Table. 6.2.

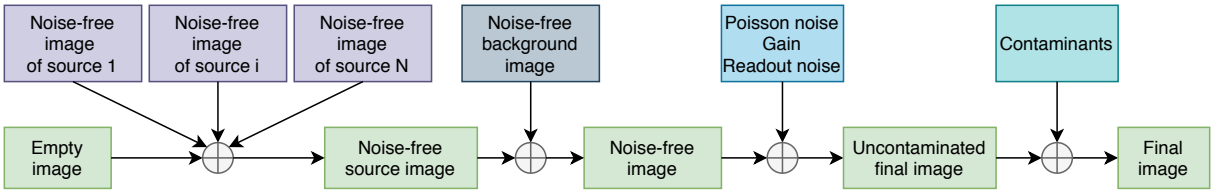


Figure 6.4: Schematic diagram of the image simulation pipeline.

Parameter	Range of values
Gain	$\mathcal{U}(0.1, 10)$ e^-/ADU
Readout noise	$\mathcal{N}(0, 5)$ in e^-
Magnitude zero point	30
Pixel size	$\mathcal{U}(0.2, 0.4)$ arcseconds
Seeing FWHM	$\mathcal{U}(0.2, 1.5)$
Background magnitude	20 mag/arcsec^2, $\pm 10\%$
Star magnitude	$\mathcal{U}(17, 26)$ or $\mathcal{U}(12, 17)$ with 1% chance
Galaxy magnitude	$\mathcal{U}(17, 22)$
Spider arm number	4 or 6
Spider arm angle	$\mathcal{U}(0, 90)$ or $\mathcal{U}(0, 60)$ degrees
Spider arm thickness	$\mathcal{U}(3, 7)$ millimeters

Table 6.2: Description of the image simulation parameters.

6.5.1 Noise-free images of isolated sources

The core of our simulations is based on noise-free images of isolated sources.

Stars

We simulate noise-free images of isolated stars with SkyMaker (Bertin, 2009).

In order to scale the image to a proper flux, a random magnitude m is picked in the range $[17, 26]$ (uniform distribution) for 99% of the sources. The faint limit has been chosen empirically so that a small but significant fraction of the stars remain virtually undetectable. For the remaining 1%, the magnitude is picked over the magnitude interval $[12, 17]$ ("bright stars"). The corresponding total flux F of the star is then computed as:

$$F = 10^{0.4(z-m)}, \tag{6.4}$$

where z is the zero point, set to 30 (which represents the typical zero-point for a unit gain detector on a professional telescope). The noise-free image $\boldsymbol{I^{(s)}}$ is finally:

$$\boldsymbol{I^{(s)}} = \frac{F}{\sum_p I_p^{(sky)}} \boldsymbol{I^{(sky)}}, \tag{6.5}$$

where $\boldsymbol{I^{(sky)}}$ is the noise-free image simulated by SkyMaker and $I^{(sky)}[p]$ denotes the pixel p of the set of pixels $\mathcal{P}$ of the image $\boldsymbol{I^{(sky)}}$. Examples of resulting noise-free images are shown in Fig. 6.5.

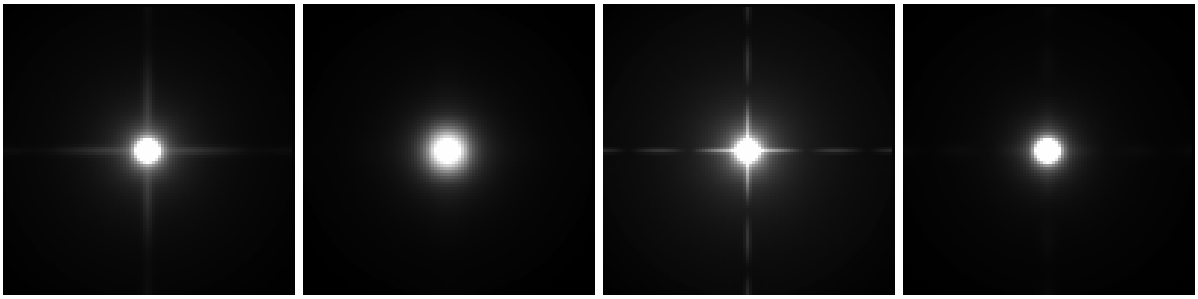


Figure 6.5: Examples of noise-free star images.

Galaxies

As we mentioned in the introduction, contrary to stars galaxies appear in a variety of shapes and sizes in astronomical images. Although analytical models exist (e.g., Sersic profiles (Sérsic, 1963)) that provide good fits to the observed light distributions of galaxies, they are far too basic to provide a good match to the images of real objects, unless they are so distant that they appear barely resolved.

There exist several compilations of real galaxy images. For instance, the EFIGI data set (Baillard et al., 2011) or the CANDELS data set (Dimauro et al., 2018), from which Boucaud et al. (2020) made a selection of 2,000 galaxies. GalSim (Rowe et al., 2015) is a software package which can be used with galaxy image thumbnails from HST observations (Mandelbaum et al., 2018) to simulate new galaxy images, as if observed with another instrument. Unfortunately, none of the solutions above offers realistic, noise-free images of isolated galaxies.

For this work we eventually chose to work with a set of 146,000 rasterized galaxy images coming from a snapshot at redshift $z = 0.5$ of a large n-body simulation (Horizon-AGN), kindly provided by Clotilde Laigle (IAP). This approach obviously has its own shortcomings. Specifically, current numerical simulations do not capture all the features found in the images of real galaxies, and the lower mass objects have a particulated aspect when viewed at high resolution. However, this represents an improvement over basic profiles at intermediate image resolutions.

Another strategy would be to simulate noise-free images of isolated galaxies with deep generative models (Lanusse et al., 2020), but this is not an option we have yet had the time to investigate.

The galaxy image simulation procedure is as follows. We convolve the high resolution galaxy images with the same PSF as the one used for stars. We then rescale the images to the simulated image pixel grid using Lanczos resampling. Beyond this point, the processing becomes very similar to that of stars, i.e., a magnitude m is picked in the range $[17, 22]$, the total flux F of the galaxy is computed as in Eq.6.4 and the final noise-free image $\boldsymbol{I^{(g)}}$ is:

$$\boldsymbol{I^{(g)}} = \frac{F}{\sum\limits_{p \in \mathcal{P}} I^{(gal)}[p]} \boldsymbol{I^{(gal)}}, \tag{6.6}$$

where $\boldsymbol{I^{(gal)}}$ is the isolated noise-free image obtained after rescaling and convolution with the PSF and $I^{(gal)}[p]$ denotes the pixel p of the set of pixels $\mathcal{P}$ of the image $\boldsymbol{I^{(gal)}}$. As with stars, the faint limit has been chosen empirically so that a small but significant fraction of the galaxies remain virtually undetectable. Examples of simulated images are shown in Fig. 6.6.

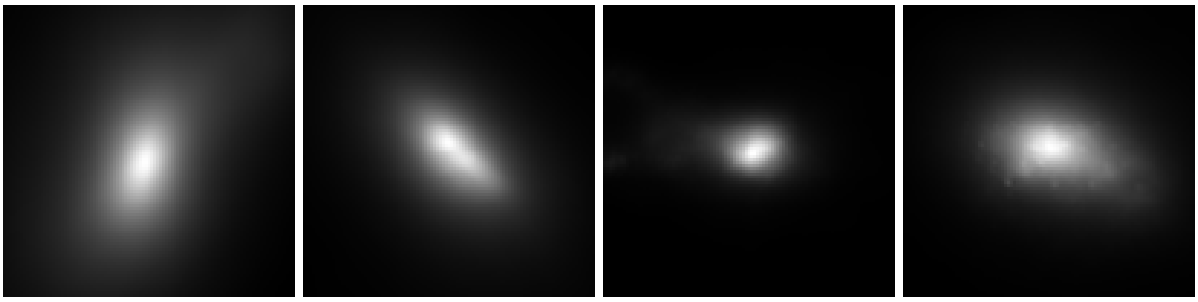


Figure 6.6: Examples of noise-free galaxy images.

For the sake of including low surface brightness objects, we also simulate another population of more diffuse "galaxies", by smoothing the raw galaxy images with a large Gaussian (15 pixel standard deviation), and zooming the result 5× compared to "regular" galaxies. We empirically limit the magnitude of such galaxies to 20 to avoid ending up with too large a number of undetectable sources.

6.5.2 Final noise-free image

In order to simulate a whole image, the noise-free images of isolated sources are added in a single image at random, independent and uniformly distributed positions. We simulate five *types* of fields, depending on the number of stars N_s and galaxies N_g added in the image:

- Type 1: low density star and galaxy fields, i.e., $(N_s, N_g) \in [\![1, 15]\!]^2$.
- Type 2: average low density fields, i.e., $N_s \in [\![20, 45]\!]$ and $N_g \in [\![1, 5]\!]$.
- Type 3: average high density, i.e., $N_s \in [\![1, 5]\!]$ and $N_g \in [\![20, 45]\!]$.
- Type 4: crowded star fields, i.e., $N_s \in [\![100, 150]\!]$ and $N_g = 0$.
- Type 5: crowded galaxy fields, i.e., $N_s = 0$ and $N_g \in [\![100, 150]\!]$.

The final noise-free image $\boldsymbol{I^{(sources)}}$ is:

$$\boldsymbol{I^{(sources)}} = \sum_{n}^{N_s} \boldsymbol{I_n^{(s)}} + \sum_{n}^{N_g} \boldsymbol{I_n^{(g)}}, \tag{6.7}$$

where $\boldsymbol{I}_{\boldsymbol{n}}^{(\boldsymbol{s})}$ is the n-th noise-free star image and $\boldsymbol{I}_{\boldsymbol{n}}^{(\boldsymbol{g})}$ is the n-th noise-free galaxy image. An example of a noise-free image is shown in Fig. 6.7.

6.5.3 Sky background flux

Once all the noise-free source images are added in a single image, a sky background corresponding to a surface brightness of magnitude 20 per square arcsecond (typical of a red band image) is added.

For 50% of the images we modulate the background by a random 3rd degree 2D-polynomial envelope with a peak-to-peak amplitude of ±10%, to simulate background gradients.

6.5.4 Noise and gain

The final processes to make a realistic image first include the Poisson noise realization:

$$\forall p \in \mathcal{P}, I^{(photons)}[p] = \text{Pois}(I^{(fluxes)}[p]), \tag{6.8}$$

where $\text{Pois}(\lambda)$ is a Poisson realization of parameter λ.

Then, a gain transformation is applied:

$$\boldsymbol{I}^{(\boldsymbol{ADU})} = \frac{1}{G}\boldsymbol{I}^{(\boldsymbol{photons})}, \tag{6.9}$$

where $G \sim \mathcal{U}(0.1, 10)$. Since the magnitude zero point is constant in ADUs, this gain transformation is equivalent to modulating the exposure time (putting aside the dynamic range).

Finally, Gaussian readout noise is added:

$$\forall p \in \mathcal{P}, I^{(f)}[p] = I^{(ADU)}[p] + R, \tag{6.10}$$

where $R \sim \mathcal{N}(0, 5)$ in e^-. An xample of final image is shown in Fig. 6.7.

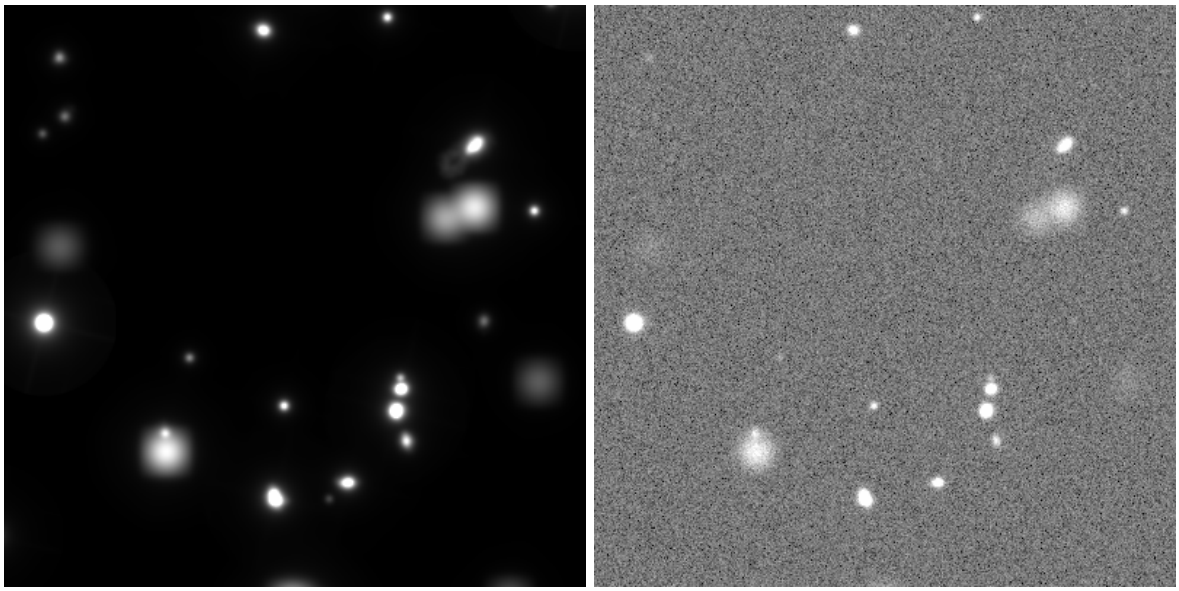


Figure 6.7: Left: noise-free source image. Right: the final image once sky background and noise are added.

6.5.5 Adding contaminants

We add a subset of contaminants to 75% of the images, so that the detector is trained to trigger only on sources that look like astronomical objects. The contaminants are:

- Bad pixels: hot and dead pixels as in the MaxiMask data set.
- Trails: as in the MaxiMask data set.
- Residual fringing patterns: as in the MaxiMask data set
- Nebulosity patterns: as in the MaxiMask data set.
- Saturation (bleed trails): each image is given a 50% chance to present saturation features, using a limited well capacity randomly selected in the range $[16000, 32000]$ e^-.. The noise-free image is simulated *without* saturation so that the source ground truth lacks the bleed trail.

Each of the contaminants above has a 50% chance to be present in a contaminated image, except for fringes and nebulosities that are mutually exclusive.

Fig. 6.8 shows examples of final images along with their ground-truth source footprints. We use a similar ground-truth representation than for MaxiMask where each scale is assigned a color: red, green, blue from the smaller to the larger. Pixels belonging to several footprints having different scales are shown with the smaller scale footprint color so that we can see the smaller objects above the larger ones.

6.5.6 Training

The CNN loss function is the sum of the softmax cross entropies of all scales. It is optimized with Adam (Kingma and Ba, 2014) and implemented in Python, using the TensorFlow library (Abadi et al., 2016). The CNN is trained for 32 epochs with $50,000$ images using a batch size of 32 (as large as possible on our GPUs). Images are dynamically compressed using the procedure introduced in Section 5.4.2. We comparing the performance on the training and testing sets, and obtain very similar numbers, which suggests that the CNN is not overfitting.

Examples of qualitative results obtained on the testing set after training are shown in Fig. 6.9. Prediction maps are built by thresholding the probability maps according to the best MC coefficient computed at the pixel level on test data.

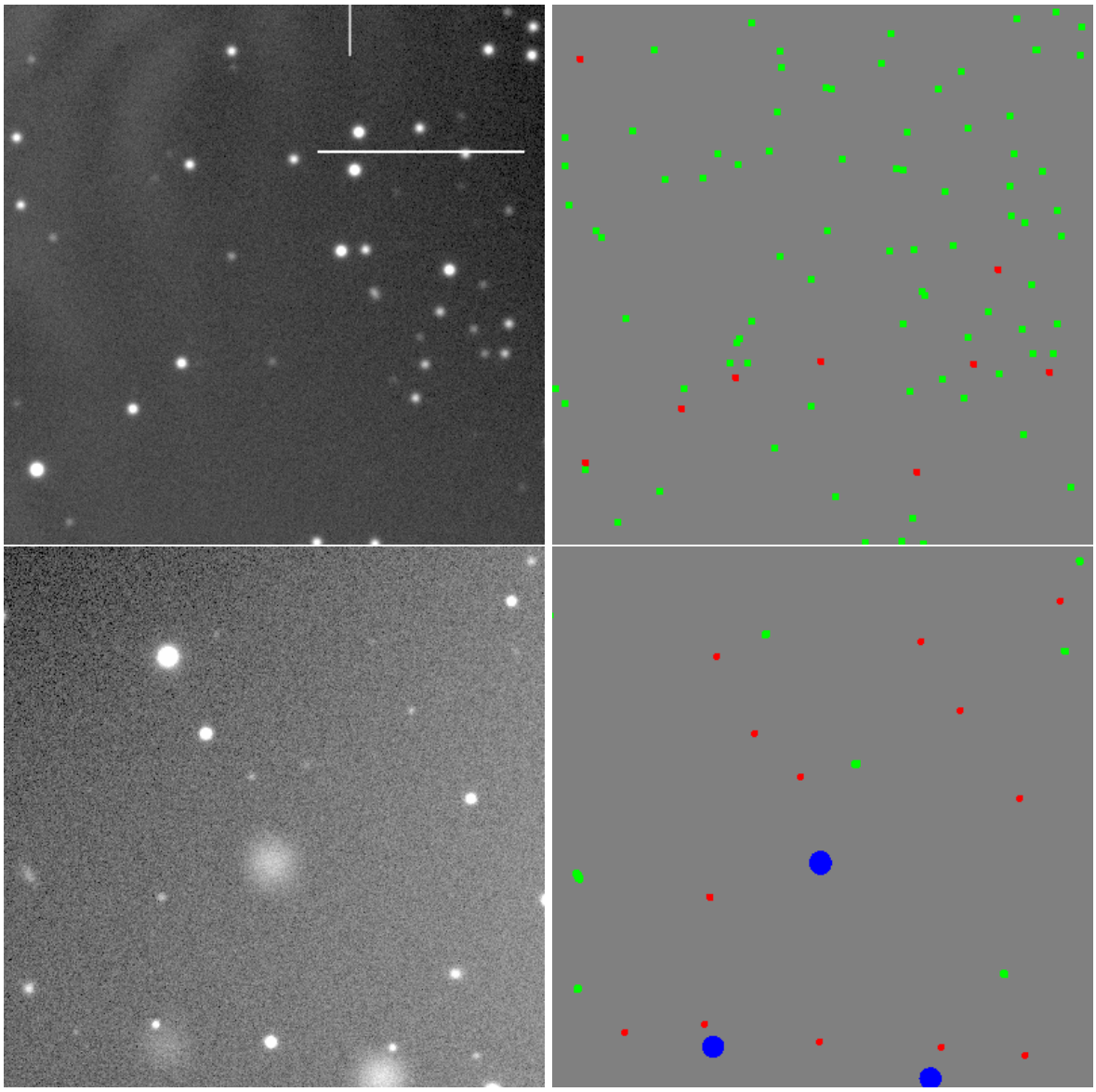


Figure 6.8: Examples of final training samples. Left: input images. Right: multiscale ground-truth footprints. Each scale is associated a color: red, green and blue from the lower to the higher scale. Pixels belonging to several footprints at different scales are shown with the color of the lower scale.

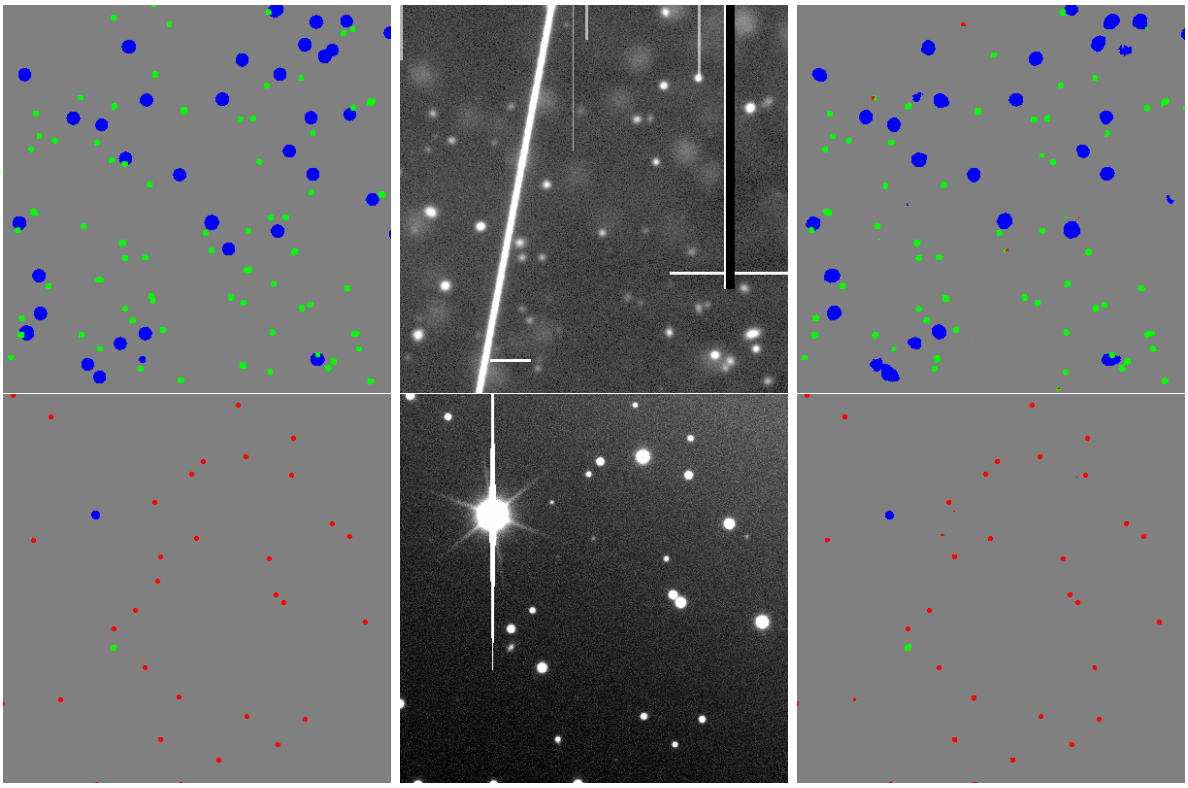


Figure 6.9: Examples of qualitative source detection results. From left to right: ground truth, input image, predictions. Color coding for the footprints is identical to that of Fig. 6.8.

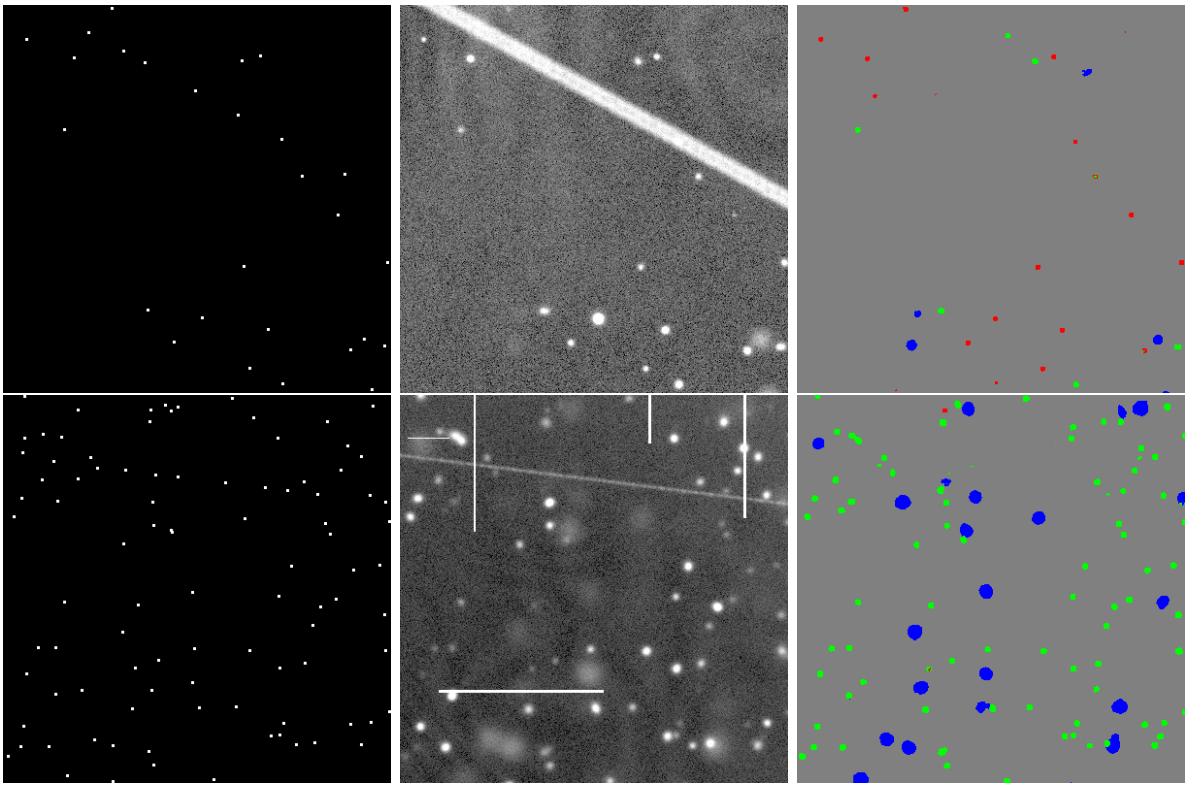


Figure 6.10: Two examples showing the main failures of SExtractor when used with default settings. SExtractor triggers false detections on contaminants and cannot detect the faintest diffuse sources. From left to right: SExtractor detections at 1.5σ with a 5×5 Gaussian filter, input image, CNN predictions.

6.6 Results

6.6.1 Qualitative comparison with SExtractor

We first qualitatively compare the CNN predictions to catalogs obtained from the same test images using the SExtractor source extraction software package (Bertin and Arnouts, 1996), run with default settings.

Fig. 6.10 shows two of the main weaknesses of SExtractor compared to the CNN in this context: it is likely to trigger false detection on contaminants and it tends to miss the faintest diffuse sources. However, SExtractor usually performs correctly on uncontaminated images.

6.6.2 Quantitative comparison with SExtractor

We examine the detection performance of the CNN and SExtractor in more details with quantitative measurements on a testing set containing $1,000$ images.

As the CNN detector performs semantic segmentation, i.e., pixel labeling, the most straightforward solution is to measure the performance at the pixel level. However, for more consistency regarding source detection and the comparison with SExtractor we decide to measure performance at the source level, using the same criterion for both detectors. Sources detected by the CNN at a given threshold are identified via connected component labeling. At each scale, we reject the connected components that are too small, using as minimum areas 1, 9 and 36 pixels for the first 3 scales.

To assess the performance of both detectors at the object level, we count the number of true positives TP, false positives FP, and false negatives FN based on the Euclidean distances between the ground truth sources and the barycenters of the predicted footprints (for the CNN), or the catalog coordinates (for SExtractor).

At every detection threshold, for every scale, we loop through all the predicted sources, matching the predicted and the ground-truth sources at the same scale if their distance is less than 3 pixels.

We then make a second pass through the data, this time by looping through every ground-truth source, and count the TPs, FPs, and FNs according to the three following cases: (1) if the ground-truth source matches one predicted source or more, one true positive is counted; (2) if the ground-truth source does not match any predicted source, one false negative is counted; (3) if during the first pass, a predicted source did not match any ground-truth source, one false positive is counted. In the later case, we check if the predicted source possibly matches a ground-truth source in an adjacent scale, in which case we consider it as a true positive.

From the TP, FP, and FN counts we derive the completeness (CP) and the contamination rate (CT), defined as:

$$\mathrm{CP} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{6.11}$$

$$\mathrm{CT} = \frac{\mathrm{FP}}{\mathrm{TP} + \mathrm{FP}}. \tag{6.12}$$

We compute CP and CT for different detection thresholds. For the CNN, the probability thresholds are chosen in the interval $[0.01, 0.99]$ every 0.01. For SExtractor, the thresholds are levels above the sky background, quantified according to the sky-background standard deviation σ_B computed by SExtractor. We use levels from $0.25\sigma_B$ to $10\sigma_B$, regularly spaced of $0.25\sigma_B$.

We compare the performance of the CNN and SExtractor in two different configurations. In the first configuration, the testing set contains only uncontaminated images. In the second configuration, the testing set contains 25% of uncontaminated images, and 75% of contaminated images, like the training set. Results are shown in Fig. 6.11.

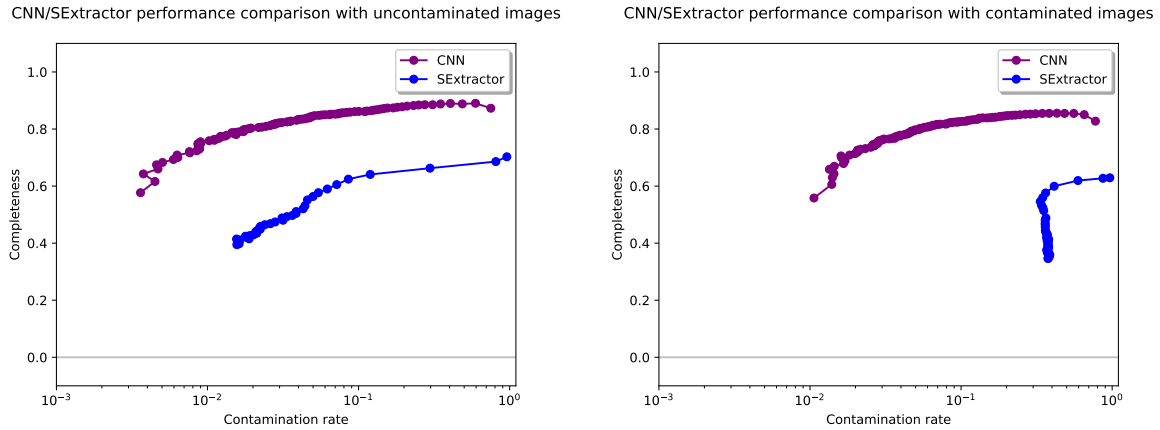


Figure 6.11: CNN and SExtractor performance comparison. Left: completeness versus contamination rate in the contaminated image regime. Right: completeness versus contamination rate in the uncontaminated regime (see text).

In the uncontaminated image regime, the CNN exhibits a $\approx 30 - 100\%$ higher completeness at a given contamination rate, compared to SExtractor. With contaminated images, the CNN clearly shows a natural robustness compared to SExtractor by reaching $20\times$ lower contamination rates. In fact the CNN performance is very similar in the two regimes.

Since completeness measurements are affected by the presence of undetectable sources, we also measure performance at multiple magnitude limits. To do this, we simply ignore sources above the given magnitude limit in the TP and FN counts. The corresponding plots are shown in Fig. 6.12.

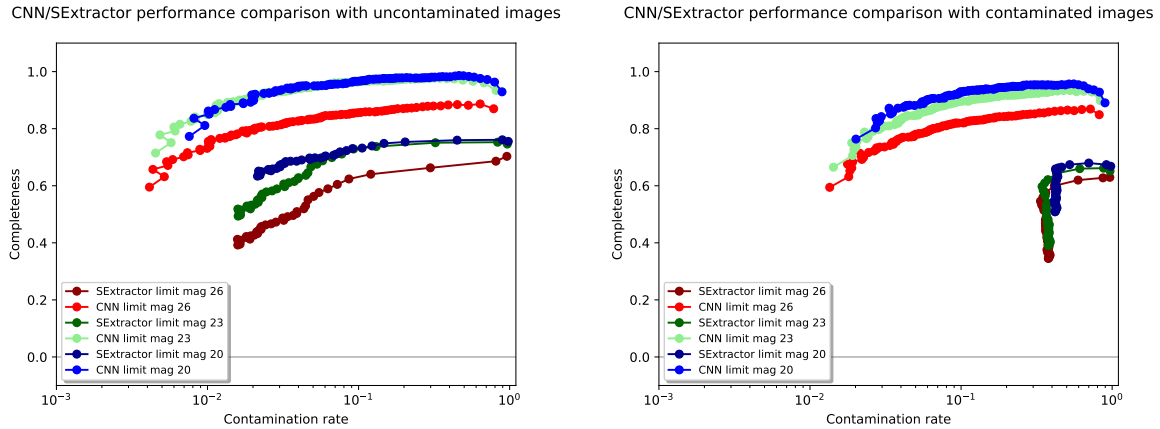


Figure 6.12: Comparison of the performance of the CNN and SExtractor at different limits in magnitude. Left: completeness versus contamination rate in the contaminated image regime. Right: completeness versus contamination rate in the uncontaminated image regime.

As expected the completeness of both detectors improves with brighter limits, although only the CNN is able to achieve near 100% completeness.

6.6.3 Qualitative test on real data

Finally, we apply the source detector to a selection of real images: a DECam exposure (Fig. 6.13) and images extracted from a much deeper r-band stack from the CFHTLS D1 field (Figs. 6.14 and

6.15). Reassuringly, we find that the detector is able to detect most sources, without triggering on trails. Some very large objects (the brightest star in Fig. 6.14 and the brightest galaxy in Fig. 6.15) are not detected, most likely because objects so large are not part of the current training set.

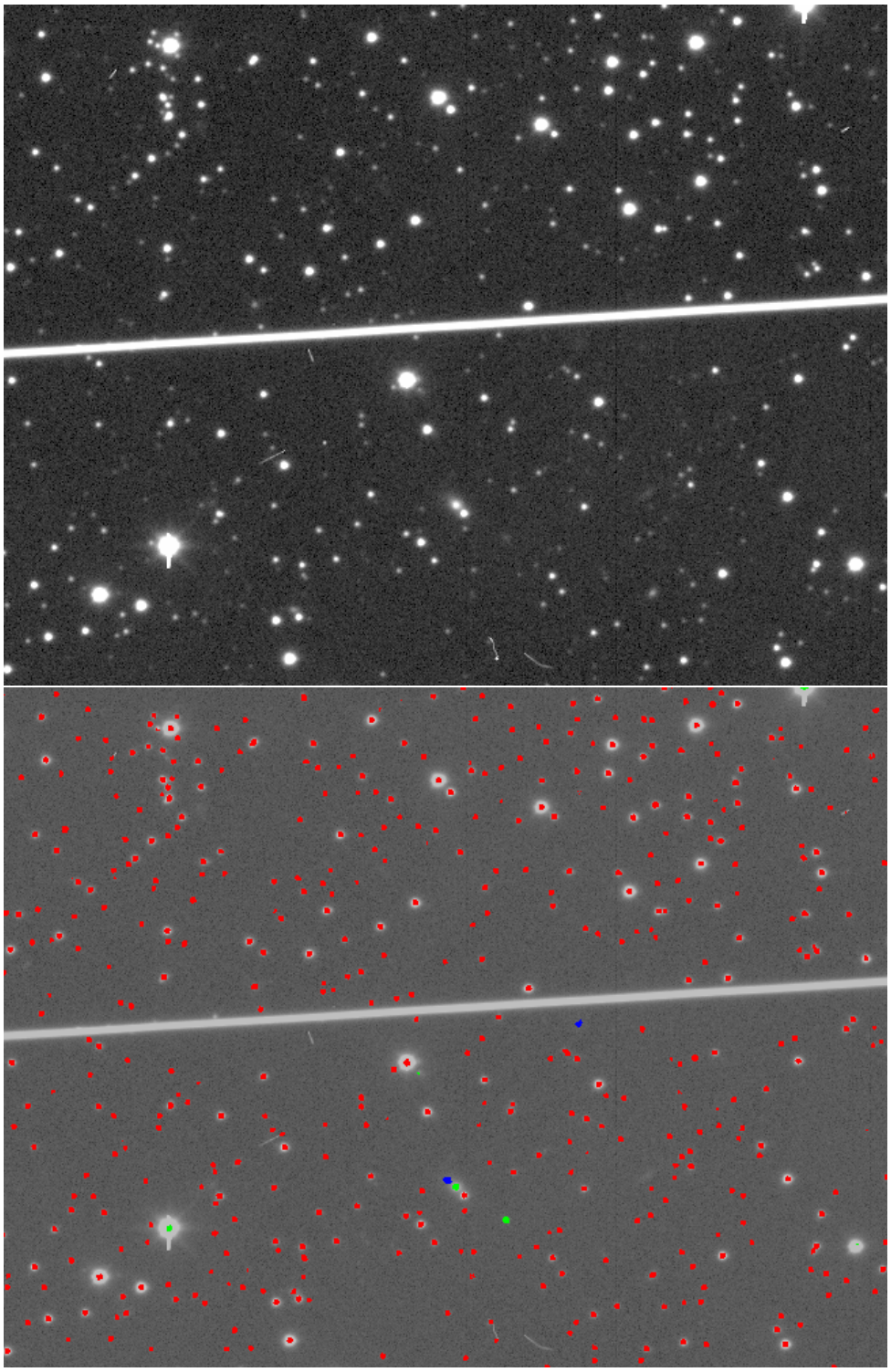


Figure 6.13: Example of a qualitative result of the CNN detector on a DECam exposure. Note how the detector does not trigger on the trail.

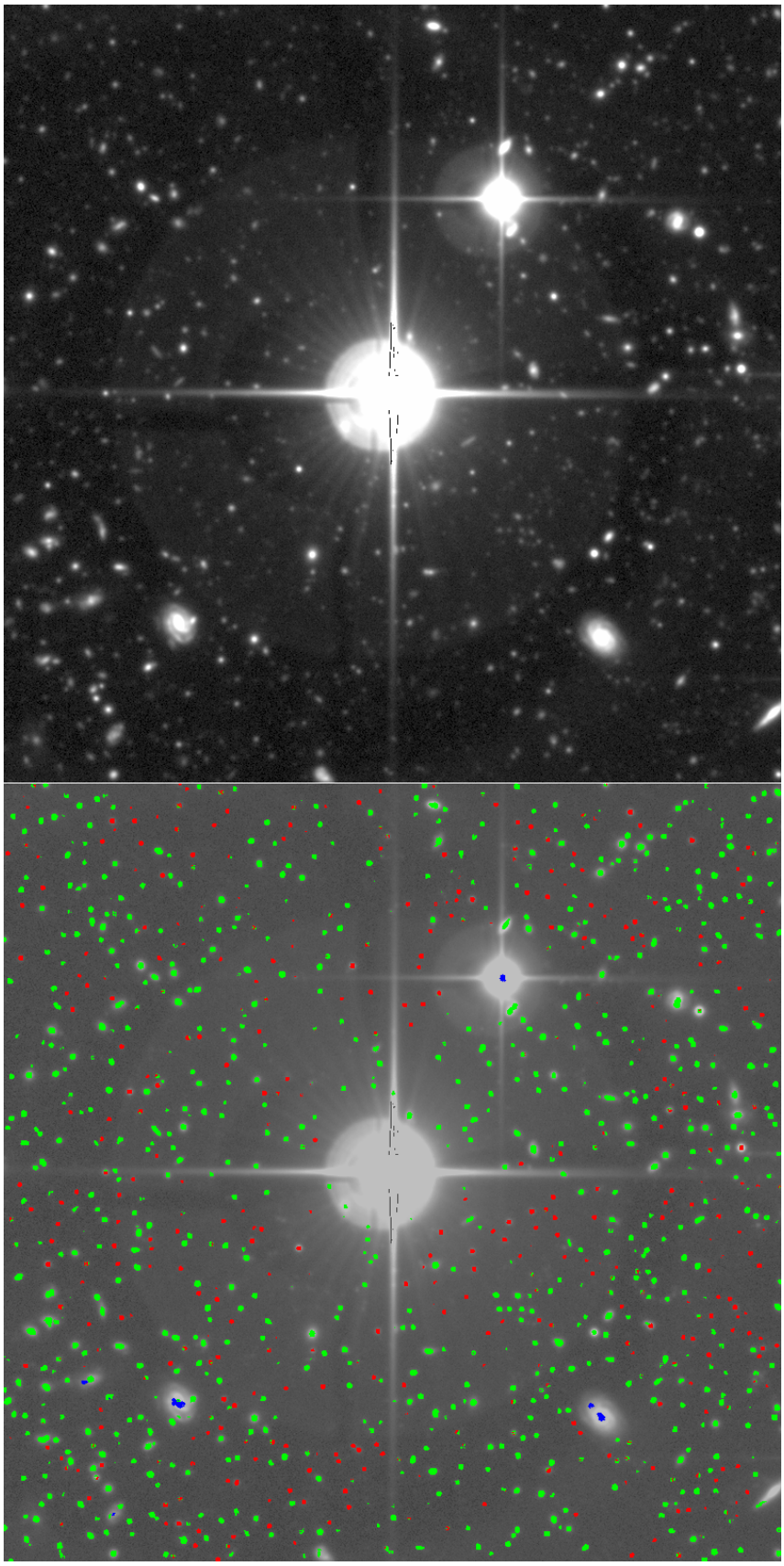


Figure 6.14: Example of a qualitative result of the CNN detector on a CFHTLS image (D1 field, r channel). Note how the detector accurately detects sources around the bright star compared to Fig. 1.1.

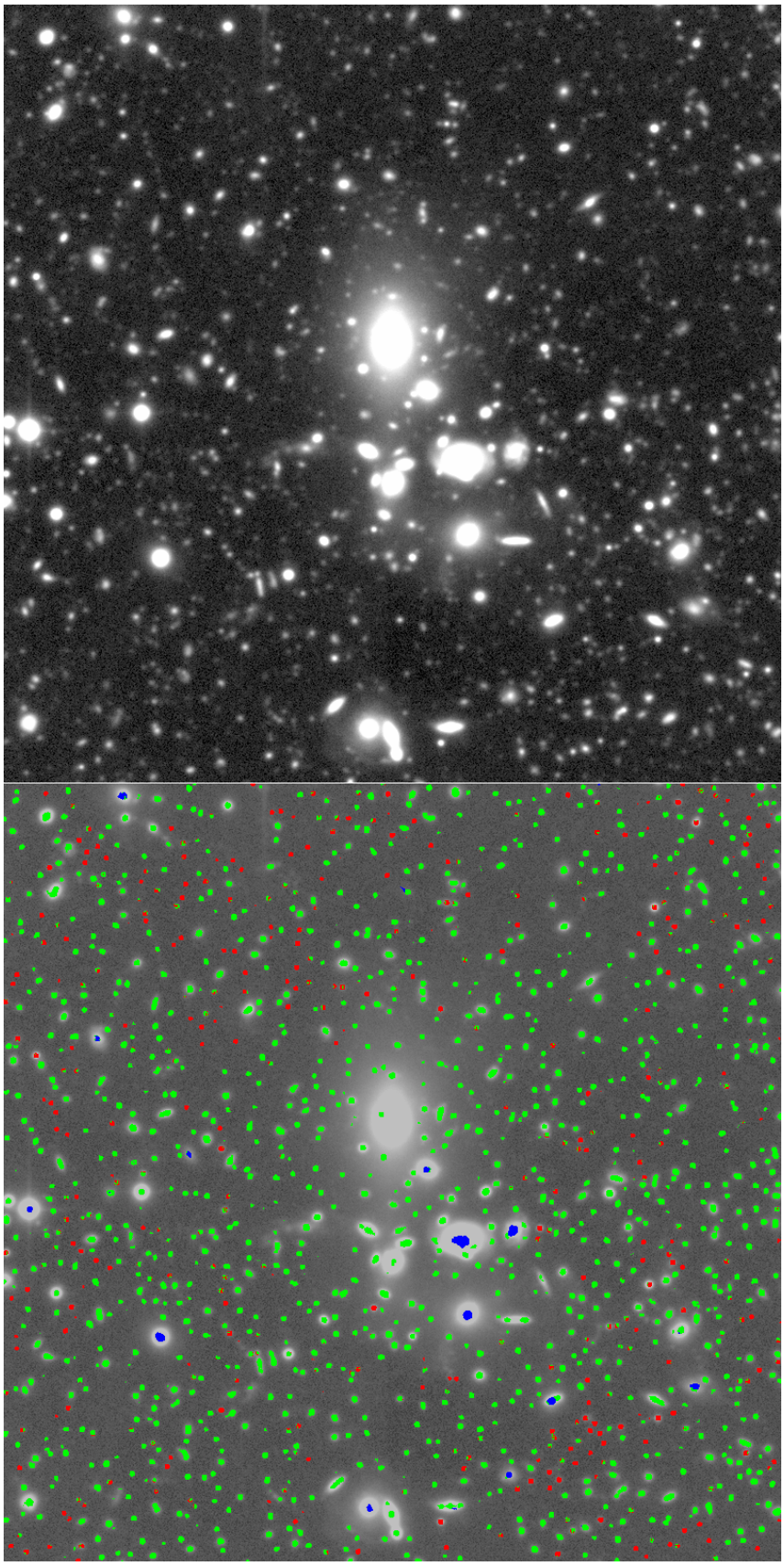


Figure 6.15: Example of a qualitative result of the CNN detector on a CFHTLS image (D1 field, r channel). Note the deblending capabilities of the detector compared to Fig. 1.1.

6.7 Conclusion and perspectives

We have designed a fully convolutional neural network for generic and robust multiscale source detection in astronomical images. The resulting detector performs semantic segmentation and identifies sources by their footprint. Using sufficiently small footprints, sources can be individually identified via connected component labeling. The multiscale aspect of the detector makes it possible to manage several object scales within the same neural network and to naturally deblend sources having different scales. In order to train our detector, we built a large training set from scratch. Starting from noise-free images of isolated sources, we were able to easily compute the ground truth (footprint and effective scale) of each source, and simulate realistic images, some of which are affected by contaminants. After training, we compared the CNN performance to a classical source extraction algorithm (SExtractor) and found that the CNN performs significantly better, especially with contaminated images. Similar to MaxiMask and MaxiTrack, the CNN assigns probabilities to every pixel, and one can adapt the prior probabilities to target different image regimes and the thresholds to target specific completeness or contamination rates. The source detector will be made available on GitHub[1] and an article is in preparation.

Although the current prototype already seems to work well with real data and to perform much better than existing algorithms, there is still room for several improvements. First of all, the source footprint criterion remains very basic and could certainly be improved. For instance, it has not yet been tested with large, "fluffy" galaxies, and is likely to generate splits in some of these objects. The scale range will also have to be revisited and extended with larger objects included in the training set.

Nevertheless the training set is certainly the main limitation of the current prototype. The distributions of stars and galaxies, as well as the galaxy shapes, are unrealistic. The next step for improving the detector and its usability for astronomical applications is therefore to build more astrophysically-minded simulations. In particular, more realistic number counts, size distribution of galaxies (e.g., Windhorst et al., 2008), and galaxy correlation functions would provide the detector with more educated priors for identifying real blends. The same goes for crowded star fields, globular clusters and star formation regions. This will require a fair amount of collaborative work. Finally, the detector could easily be extended to manage multispectral images. However, while multiple bands would provide additional information to the CNN, this would also make the detector much more instrument dependent, and the training set would have to be tuned accordingly.

[1] https://github.com/mpaillassa

Chapter 7

Conclusion

In this thesis, we designed new algorithms to extract more reliable catalogs from astronomical images. Taking advantage of deep learning and convolutional neural network techniques (LeCun et al., 1995; Krizhevsky et al., 2012; Badrinarayanan et al., 2017), we developed state-of-the-art models that can readily be applied to a broad range of optical and NIR wide-field images, in a fully automated way.

This work was largely data driven, and is illustrative of a new approach to develop data processing tools. Indeed, data-driven models mostly rely on carefully designed training data sets and training procedures. Data sets must be large enough and representative of the task that must be solved to provide generalization, i.e., the ability of the trained models to perform well on new data. During the duration of the thesis, more than 50 TB of heterogeneous image data were processed, either by custom data analysis programs or through training procedures.

Our first realizations, MaxiMask and MaxiTrack (Paillassa et al., 2020), are contaminant detectors. MaxiMask and MaxiTrack allow for a large variety of image defects and exposures to be flagged, and can be used in complement to traditional source detection algorithms or for automated image quality control. For training MaxiMask and MaxiTrack CNNs, we built realistic data sets covering a wide diversity of images retrieved from various instruments of the COSMIC-DANCE survey, as well as image simulations.

Facing strong class imbalance issues with astronomical image pixels, we defined an empirical cost weighting strategy and proposed a rescaling scheme of the detector outputs based on a Bayesian approach. This approach allows different image regimes to be managed by simply updating a set of priors, and appears to work well in practice.

Our analysis of MaxiMask inferences showed that it generalizes well on real data, including data originating from instruments not used for training. MaxiMask and MaxiTrack are publicly available for inference[1].

We are aware that contaminants particularly harmful to wide-field images, such as optical ghosts, reflections and scattered light are not yet taken into account by MaxiMask. Since they are difficult to simulate or to isolate from images, building a training set requires a significant amount of work, similar to what had to be done for diffraction spikes. This is an endeavor that I am ready to undertake in the near future, as I will be working on HSC images as part of my post-doctoral studies. Provided additional data is incorporated into the training set, other serious contaminants such as crosstalks could also be taken into account by MaxiMask, while performance on existing contaminants could be improved, especially for diffraction spikes.

Our second realization is a robust multiscale detector for astronomical sources. Our detector is able to deal with various issues that hamper traditional source detection algorithms such as extended, low surface brightness objects, source blending, and contamination by image defects.

[1] https://github.com/mpaillassa/MaxiMask

The multiscale aspect of our approach allows the source detector to recover sources at different scales in separate output maps. We assessed the performance of the new detector on test data and found a $\approx 30 - 100\%$ higher completeness, and a $20\times$ lower contamination rate compared to a traditional source extraction algorithm (SExtractor). A paper is in preparation and the CNN source detector will be available for inference on GitHub.

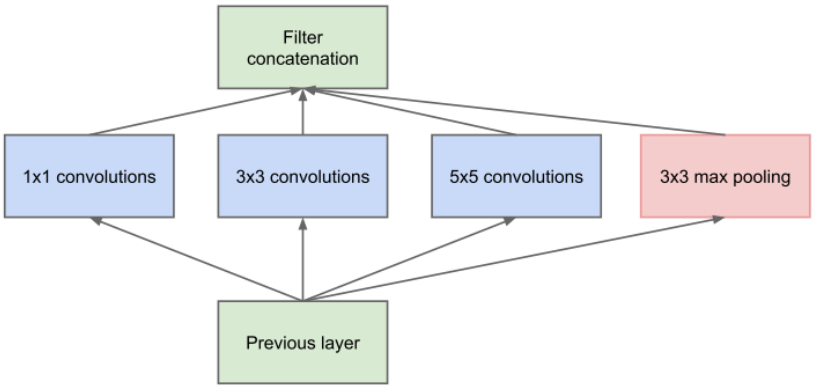
Work is also ongoing to implement it in the SourceXtractor++ source package (Bertin et al., 2019b), which will be extensively used to analyze imaging data from a variety of sources in the context of the Euclid mission. The output footprint maps of the CNN detector are similar to the segmentation map produced by the current SourceXtractor++ detector so that they can almost be used as a drop-in replacement, and will provide initial guesses for the source fitting procedures.

Although the performance of the current CNN detector is already satisfying, there is room for improvements at the data level. In particular, we are aware that our galaxy images and their distribution are unrealistic, which necessarily impacts the behavior of the detector. More work is clearly needed to include astrophysical models in the simulations. Additionally, and although it is not our primary goal, specific training sets could also be built for targeted instruments or scientific goals. Finally, we note that catalogs produced by a multiscale, highly robust and adaptive detector such as ours potentially have a more complex selection function than, e.g., surface brightness or magnitude-limited catalogs produced by simpler algorithms. Simulated image datasets will also be necessary at this level for deriving the selection functions and taking full advantage of the new catalogs.

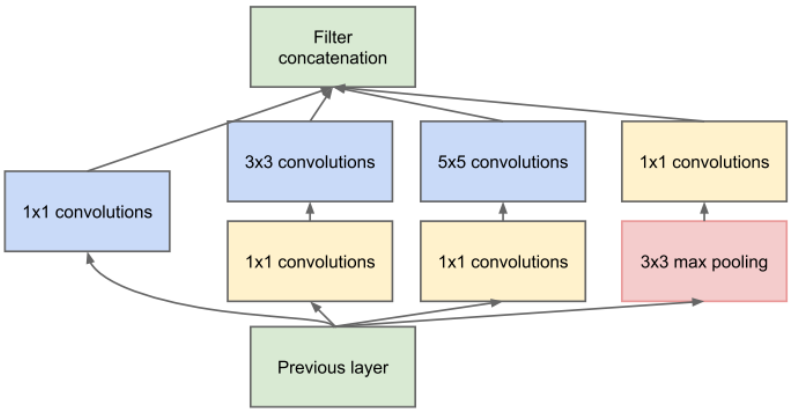
Appendices

Appendix A

Classical CNN architectures for image classification



(a) Inception module, naïve version



(b) Inception module with dimensionality reduction

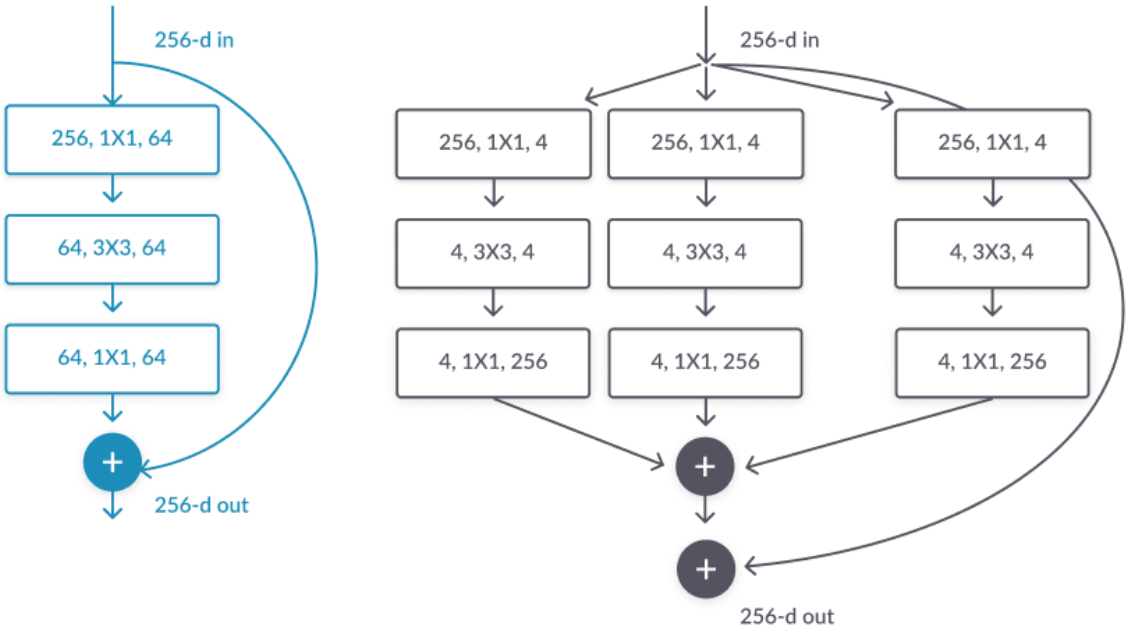


Figure A.1: Top: the two versions of Inception blocks in GoogLeNet. Image credit: Szegedy et al. (2015). Bottom: Resnet (He et al., 2016) and ResNeXt (Xie et al., 2017) residual blocks. Image credit: https://missinglink.ai/guides/keras/.

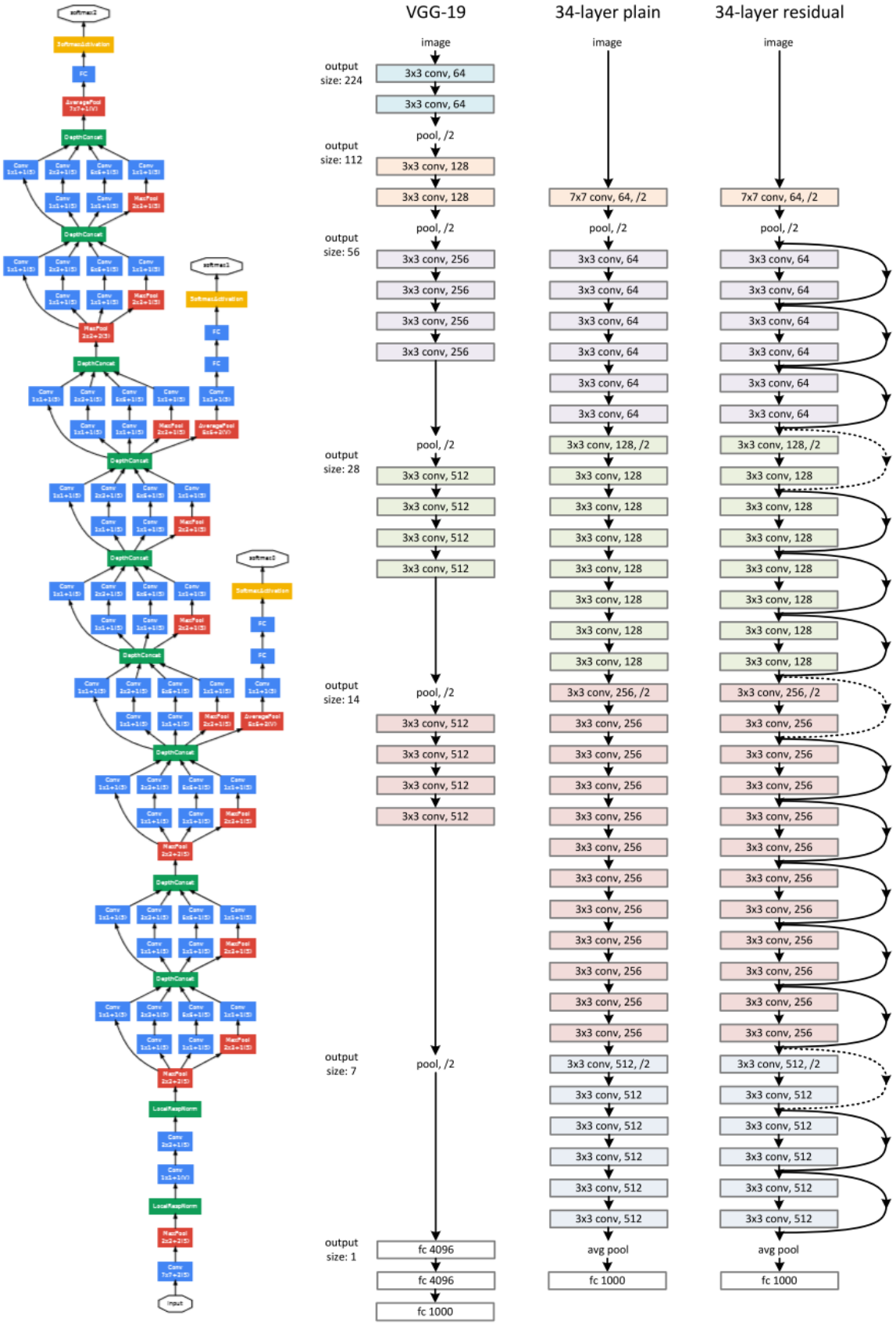
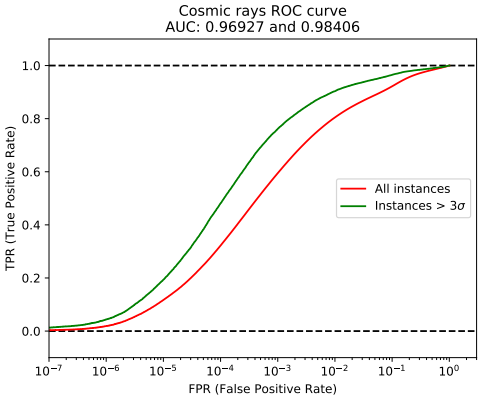
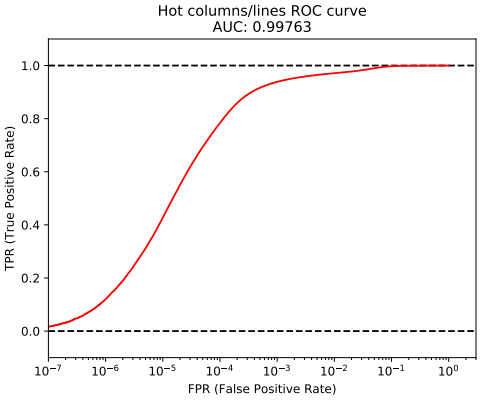


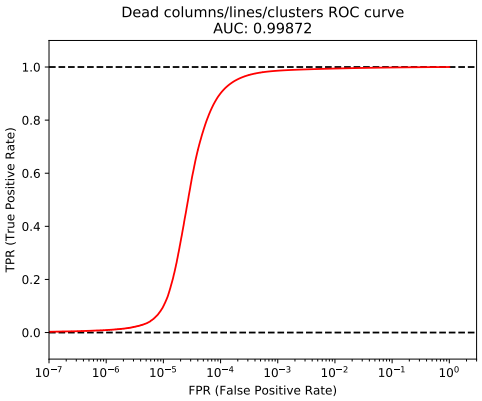
Figure A.2: Overall representations of GoogLeNet (Szegedy et al., 2015), VGG-19 (Simonyan and Zisserman, 2014) and ResNet (He et al., 2016) architectures. Images credits: Szegedy et al. (2015) and He et al. (2016).

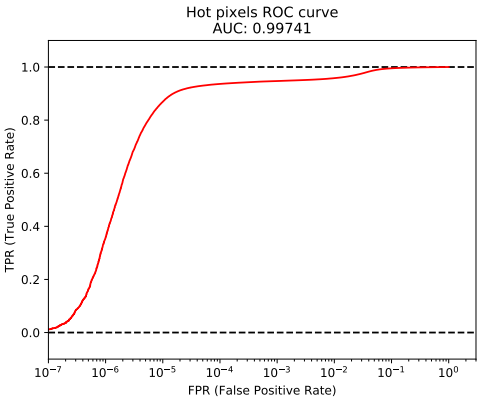
Appendix B

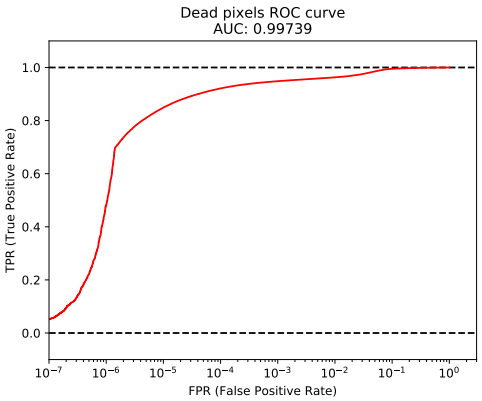
MaxiMask performance curves

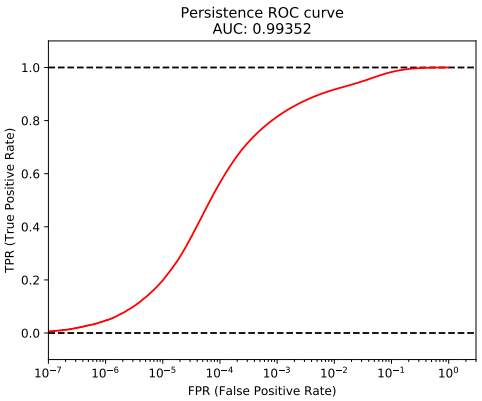

Cosmic rays ROC curve
AUC: 0.96927 and 0.98406
All instances
Instances > 3σ
TPR (True Positive Rate)
FPR (False Positive Rate)

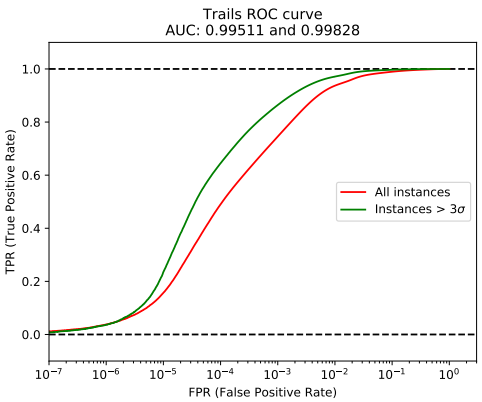

Hot columns/lines ROC curve
AUC: 0.99763
TPR (True Positive Rate)
FPR (False Positive Rate)

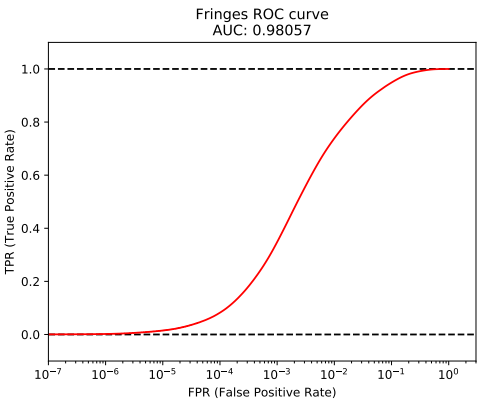

Dead columns/lines/clusters ROC curve
AUC: 0.99872
TPR (True Positive Rate)
FPR (False Positive Rate)


Hot pixels ROC curve
AUC: 0.99741
TPR (True Positive Rate)
FPR (False Positive Rate)


Dead pixels ROC curve
AUC: 0.99739
TPR (True Positive Rate)
FPR (False Positive Rate)


Persistence ROC curve
AUC: 0.99352
TPR (True Positive Rate)
FPR (False Positive Rate)


Trails ROC curve
AUC: 0.99511 and 0.99828
All instances
Instances > 3σ
TPR (True Positive Rate)
FPR (False Positive Rate)


Fringes ROC curve
AUC: 0.98057
TPR (True Positive Rate)
FPR (False Positive Rate)

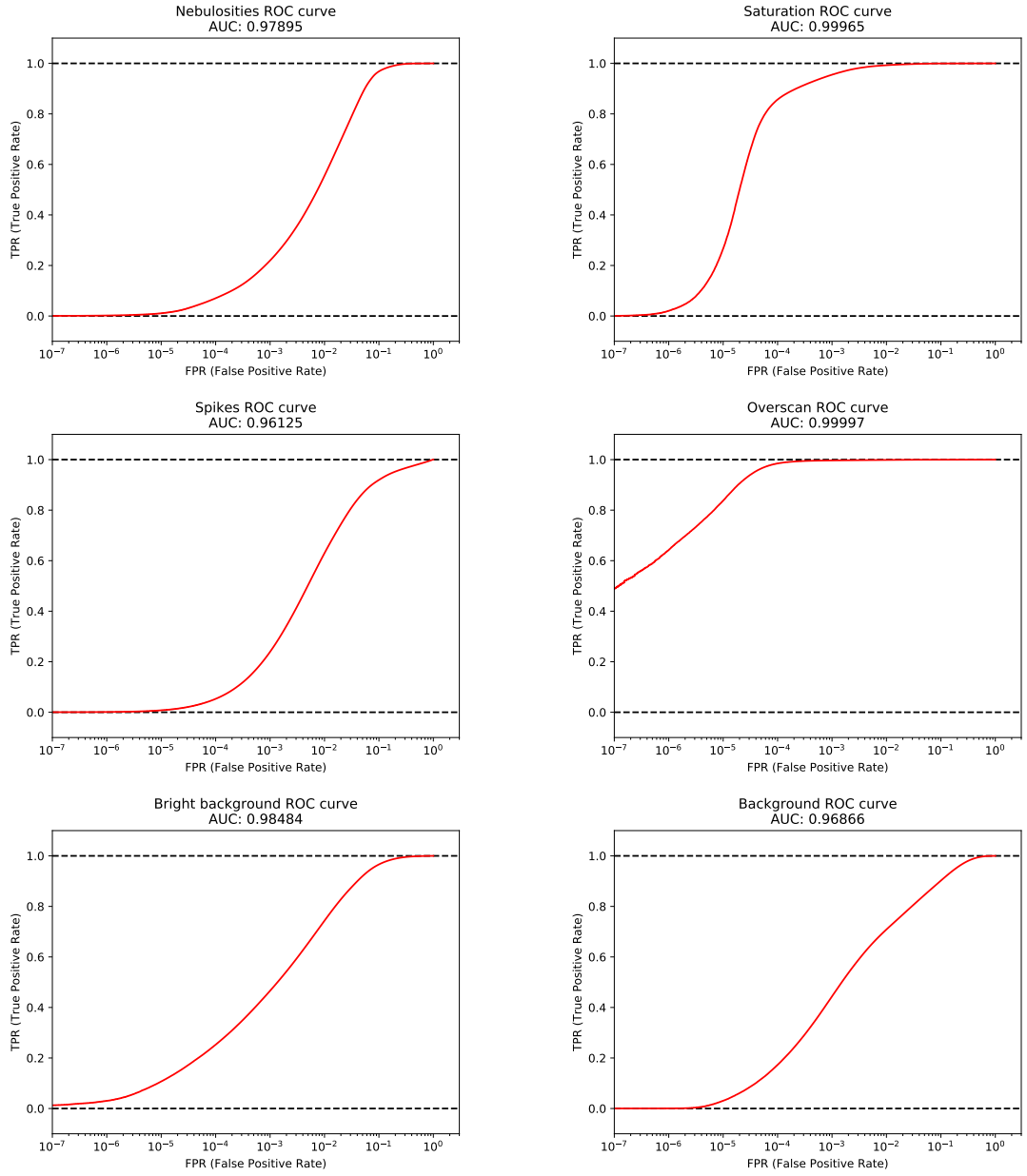
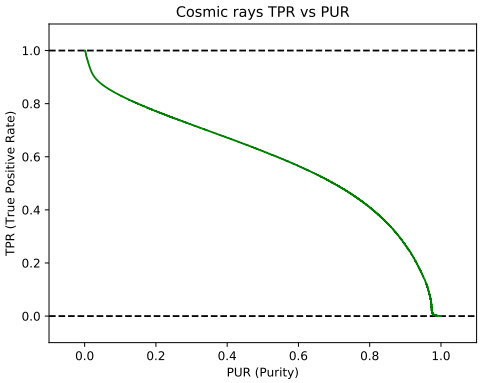
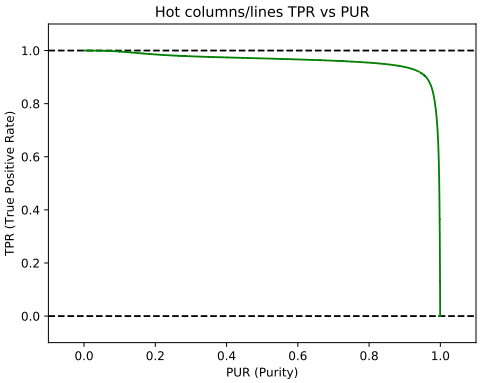
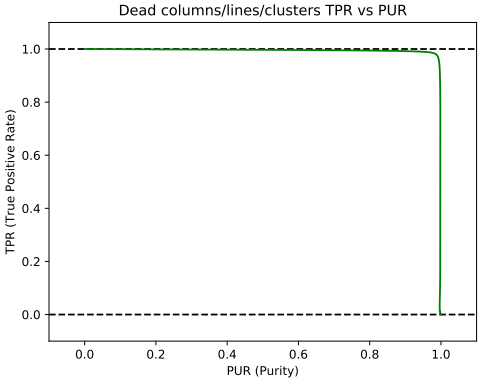
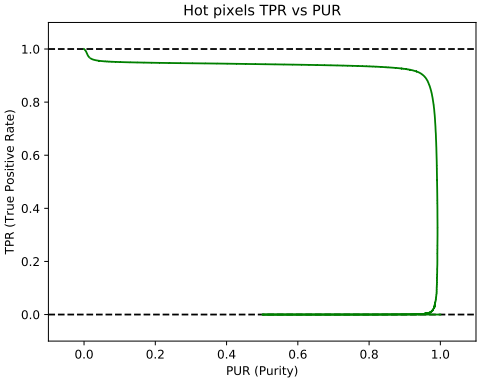


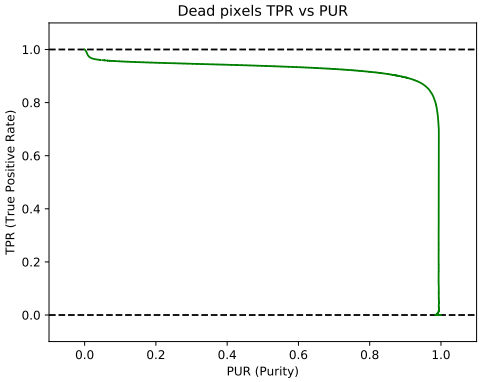
Figure B.1: ROC curves: TPR vs FPR. The FPR axis in in logarithmic scale so that very low FPR are best visualized. The ROC curve and the AUC are provided for each class. Images credit: Paillassa et al. (2020).

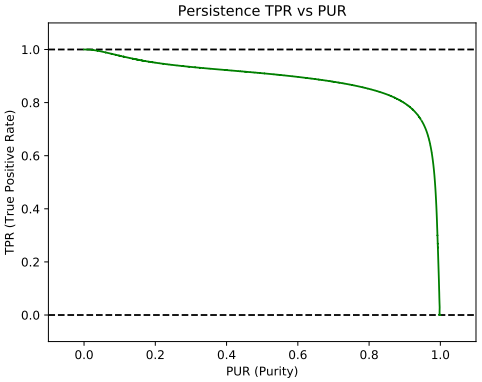

Cosmic rays TPR vs PUR
TPR (True Positive Rate)
PUR (Purity)
0.0
0.2
0.4
0.6
0.8
1.0

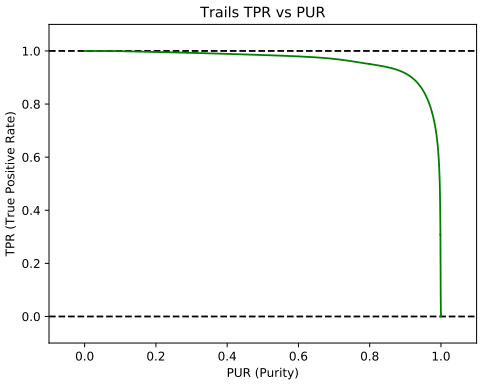

Hot columns/lines TPR vs PUR
TPR (True Positive Rate)
PUR (Purity)
0.0
0.2
0.4
0.6
0.8
1.0

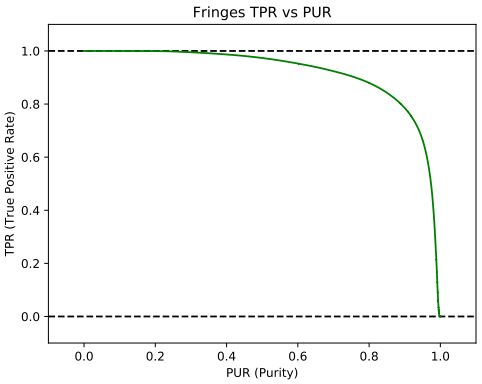

Dead columns/lines/clusters TPR vs PUR
TPR (True Positive Rate)
PUR (Purity)
0.0
0.2
0.4
0.6
0.8
1.0


Hot pixels TPR vs PUR
TPR (True Positive Rate)
PUR (Purity)
0.0
0.2
0.4
0.6
0.8
1.0


Dead pixels TPR vs PUR
TPR (True Positive Rate)
PUR (Purity)
0.0
0.2
0.4
0.6
0.8
1.0


Persistence TPR vs PUR
TPR (True Positive Rate)
PUR (Purity)
0.0
0.2
0.4
0.6
0.8
1.0


Trails TPR vs PUR
TPR (True Positive Rate)
PUR (Purity)
0.0
0.2
0.4
0.6
0.8
1.0


Fringes TPR vs PUR
TPR (True Positive Rate)
PUR (Purity)
0.0
0.2
0.4
0.6
0.8
1.0

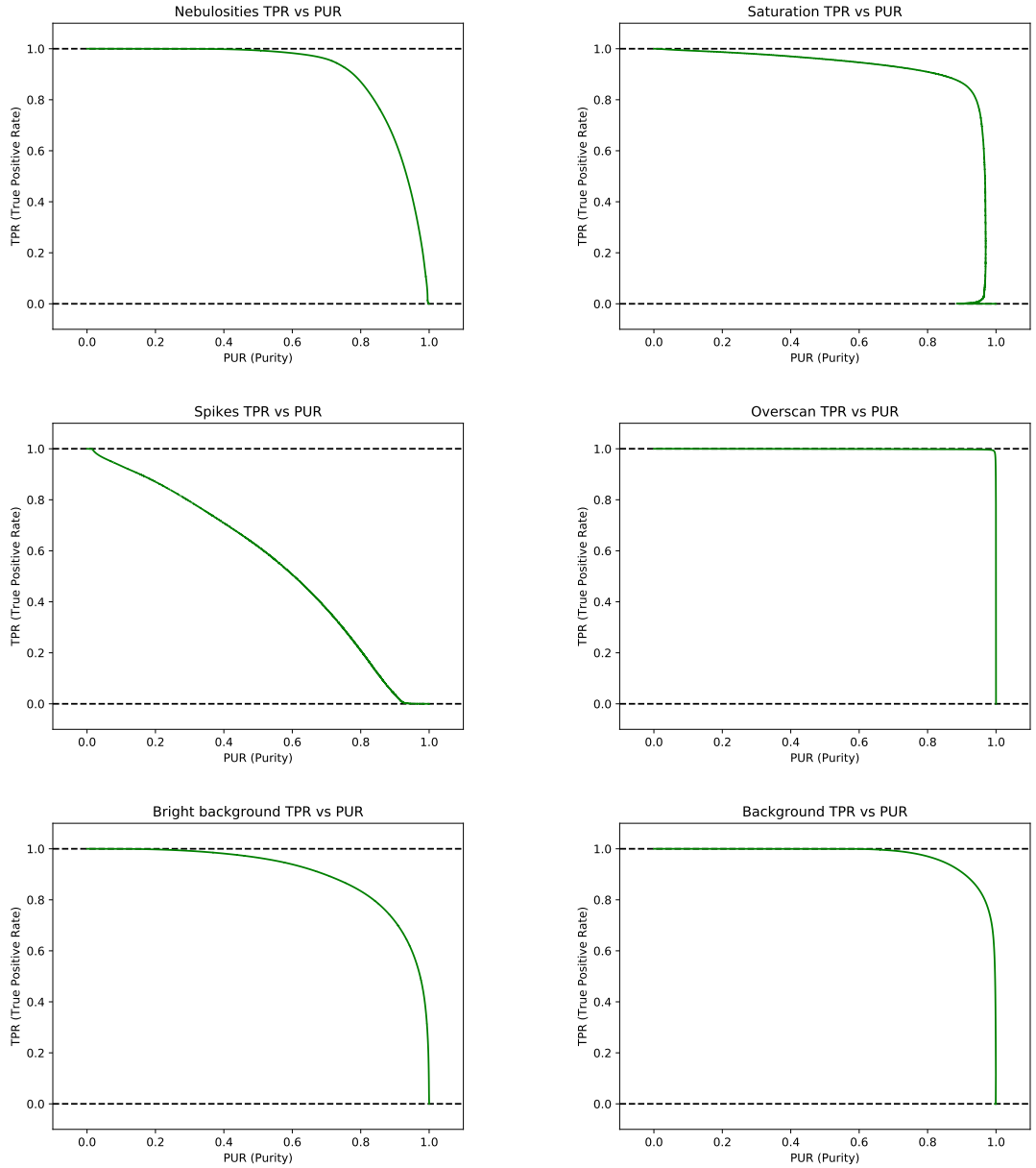
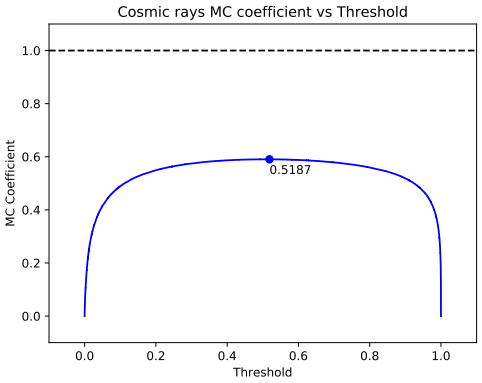
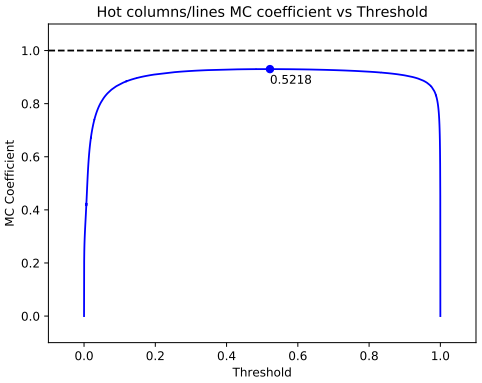
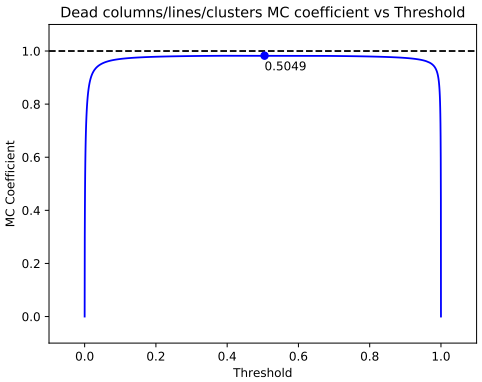
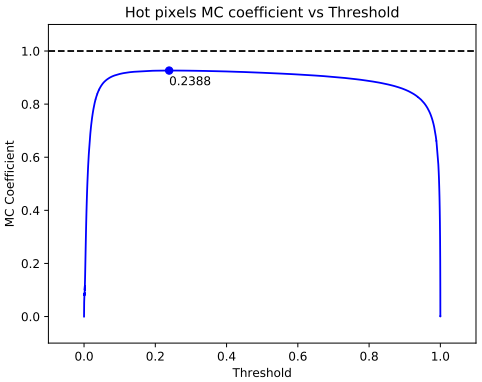


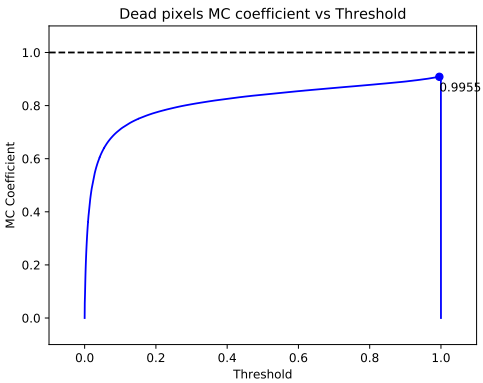
Figure B.2: Purity curves: TPR vs PUR. Images credit: Paillassa et al. (2020).

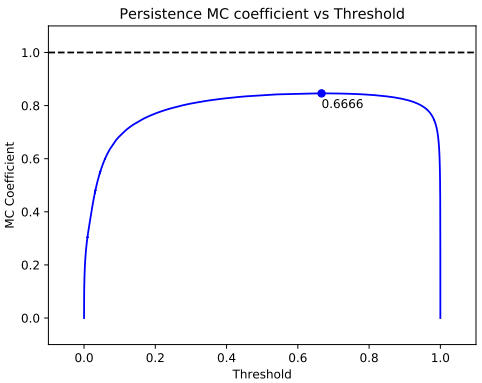

Cosmic rays MC coefficient vs Threshold
MC Coefficient
Threshold
0.5187

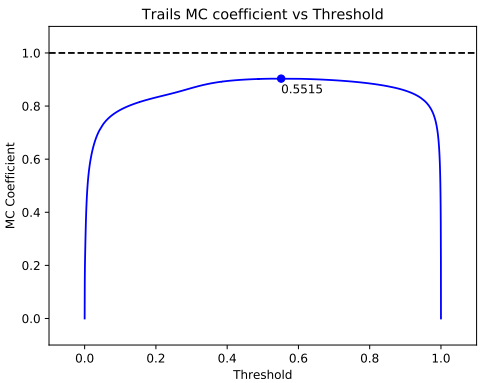

Hot columns/lines MC coefficient vs Threshold
MC Coefficient
Threshold
0.5218

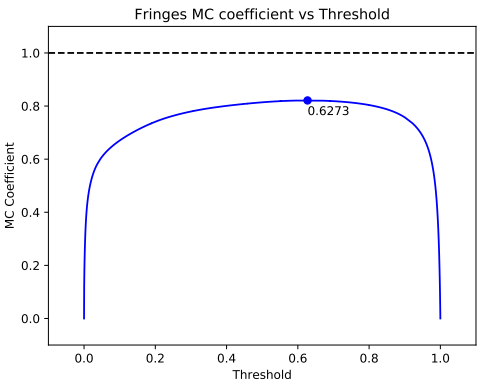

Dead columns/lines/clusters MC coefficient vs Threshold
MC Coefficient
Threshold
0.5049


Hot pixels MC coefficient vs Threshold
MC Coefficient
Threshold
0.2388


Dead pixels MC coefficient vs Threshold
MC Coefficient
Threshold
0.9955


Persistence MC coefficient vs Threshold
MC Coefficient
Threshold
0.6666


Trails MC coefficient vs Threshold
MC Coefficient
Threshold
0.5515


Fringes MC coefficient vs Threshold
MC Coefficient
Threshold
0.6273

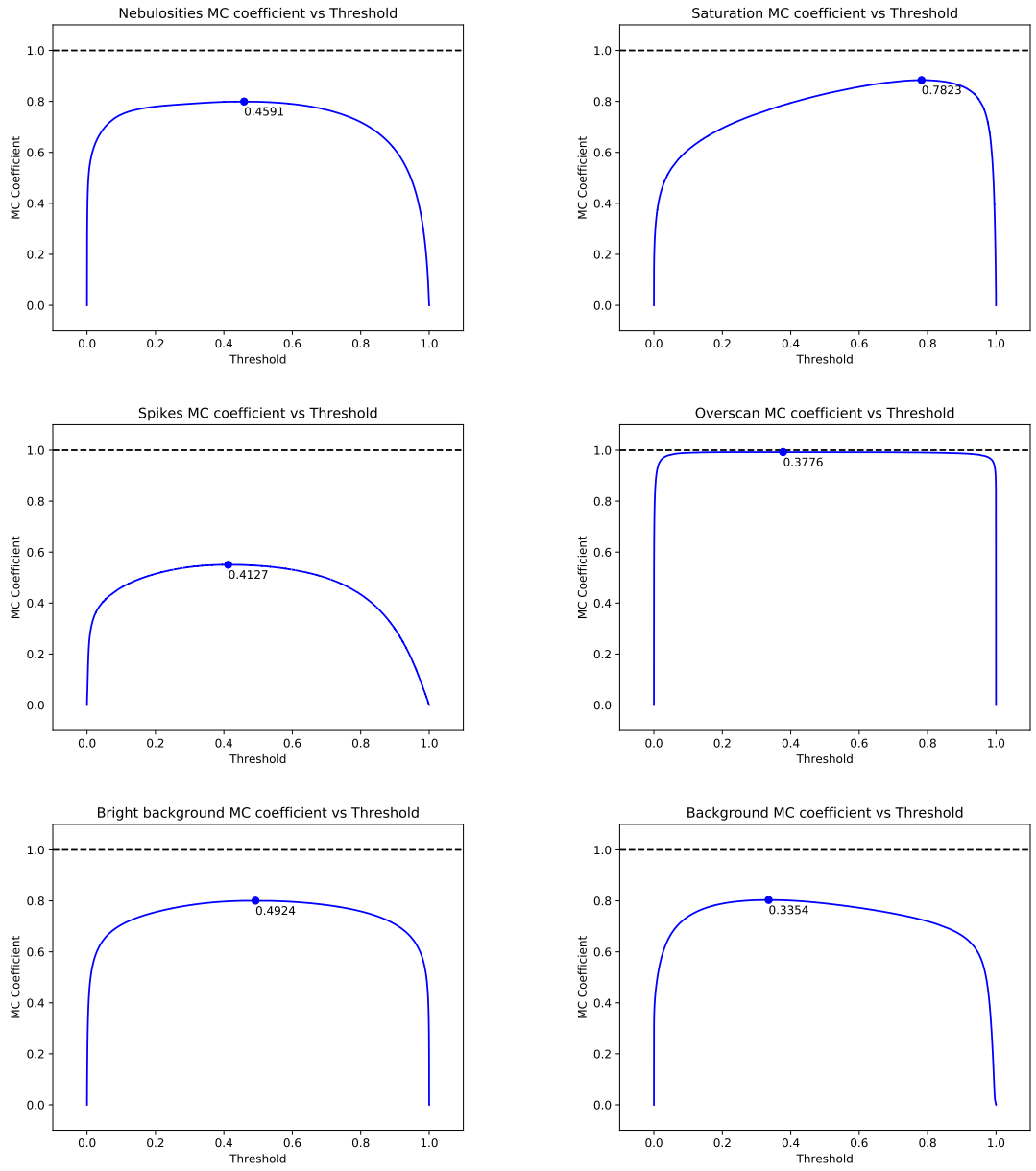
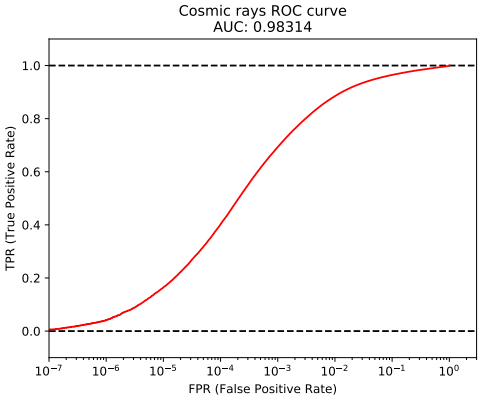
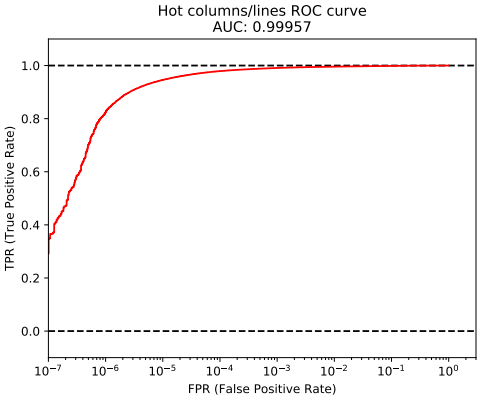
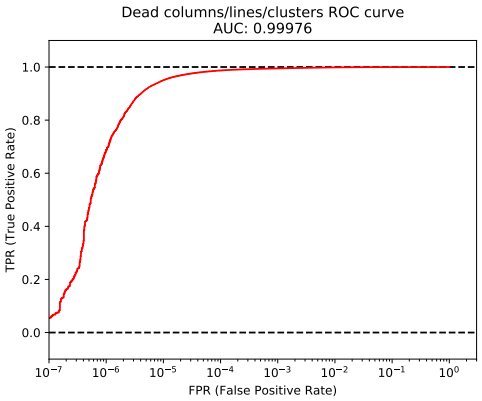
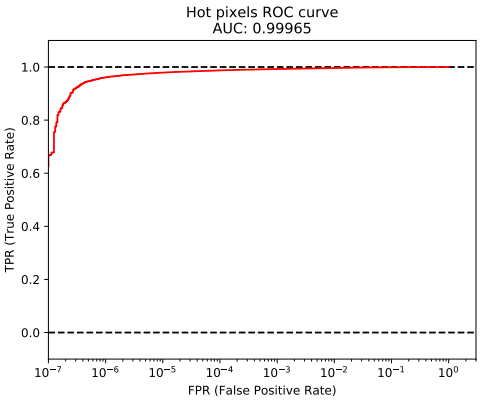


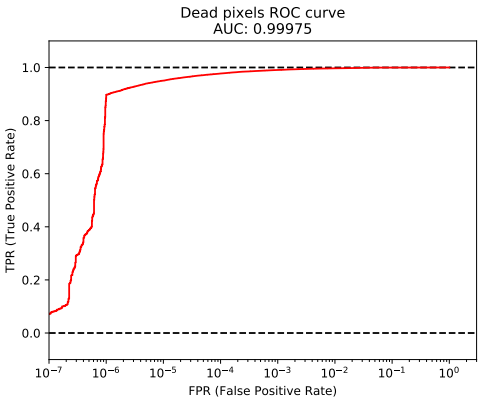
Figure B.3: MC coefficient curves: MC coefficient vs Detection threshold. On each curve is annotated the threshold for which the MC coefficient is the highest. These curves were computed using the probabilities corrected from priors using empirical training priors. Images credit: Paillassa et al. (2020).

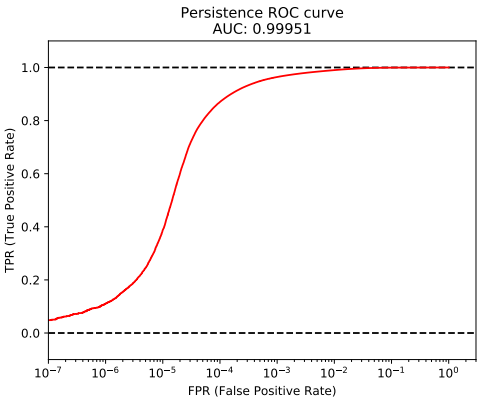

Cosmic rays ROC curve
AUC: 0.98314
TPR (True Positive Rate)
FPR (False Positive Rate)

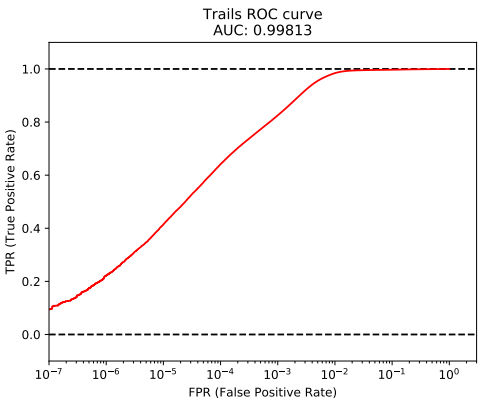

Hot columns/lines ROC curve
AUC: 0.99957
TPR (True Positive Rate)
FPR (False Positive Rate)

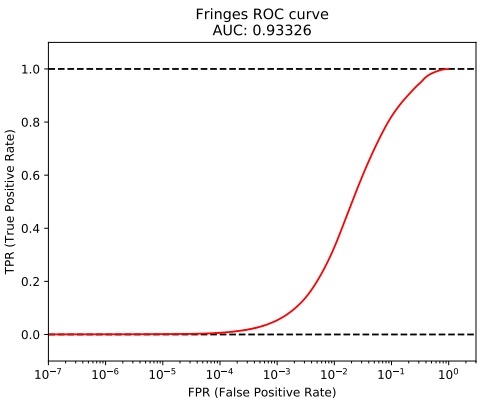

Dead columns/lines/clusters ROC curve
AUC: 0.99976
TPR (True Positive Rate)
FPR (False Positive Rate)


Hot pixels ROC curve
AUC: 0.99965
TPR (True Positive Rate)
FPR (False Positive Rate)


Dead pixels ROC curve
AUC: 0.99975
TPR (True Positive Rate)
FPR (False Positive Rate)


Persistence ROC curve
AUC: 0.99951
TPR (True Positive Rate)
FPR (False Positive Rate)


Trails ROC curve
AUC: 0.99813
TPR (True Positive Rate)
FPR (False Positive Rate)


Fringes ROC curve
AUC: 0.93326
TPR (True Positive Rate)
FPR (False Positive Rate)

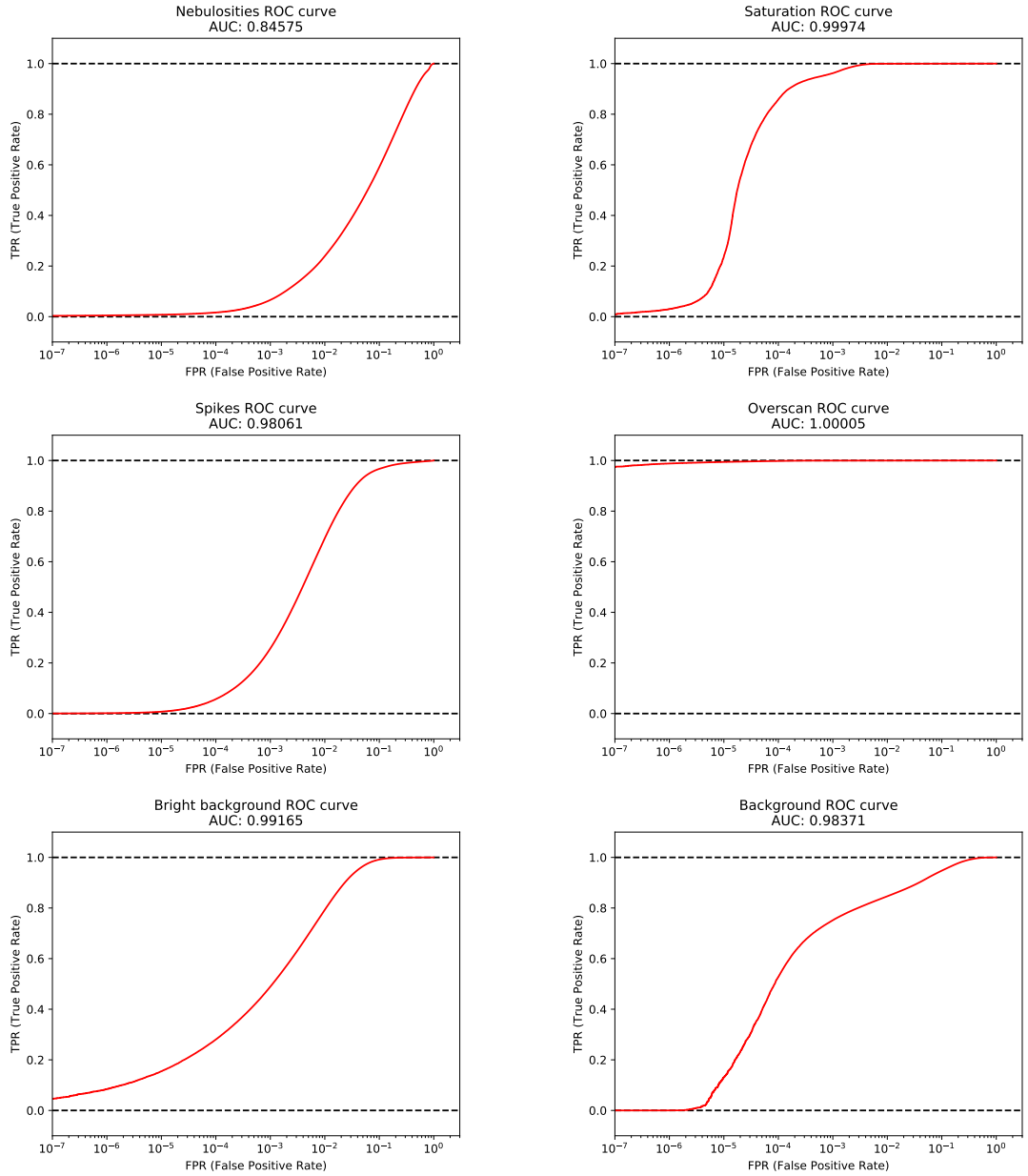


Figure B.4: ROC curves computed on the context robustness test sets from Section 5.6.1.

Appendix C

Introduction en français

Une grande partie de la science menée en Astrophysique dépend des catalogues de sources. La plupart des sources astronomiques cataloguées à ce jour ont été détectées dans des images grand champ prises dans les longueurs d'onde visible et proche infrarouge (PIR). La détection de sources est donc une étape cruciale dans l'exploitation des données d'imagerie, en particulier dans les grands relevés photométriques du ciel. Cependant, les performances de détection actuelles en termes de fiabilité et de complétude sont désormais insuffisantes au regard des exigences scientifiques des expériences en cours et à venir, comme par exemple, HSC (Aihara et al., 2018), Euclid (Racca et al., 2016), ou le LSST (Ivezić et al., 2019). Un gain en performance est nécessaire, en prenant en compte les contraintes de temps imposées par la grande quantité de données à traiter.

Dans ce contexte, notre but est de concevoir un détecteur de source le plus universel possible pour les instruments grand champ dans les domaines optique et PIR. Par universel, nous entendons qu'il doit être capable de fonctionner avec des images provenant de divers télescopes, caméras et conditions d'observations, sans nécessiter de réglages importants. Nous visons également à réaliser un détecteur robuste vis-à-vis des défauts ou imperfections pouvant affecter les images.

Ce projet a été initié dans le cadre de deux relevés en particulier : Cosmic-DANCe (Bouy et al., 2017), pour *Dynamical Analysis of Near Clusters*, ci-après écrit COSMIC-DANCE, et Euclid (Laureijs et al., 2012).

Le but principal du relevé COSMIC-DANCE est de retrouver la fonction de masse stellaire initiale, c'est-à-dire la fonction décrivant le taux de formation des étoiles en fonction de leur masse, en étudiant les amas ouverts jeunes et proches. COSMIC-DANCE se concentre en particulier sur les étoiles de faible masse, allant au-dessous de la limite de magnitude de la mission Gaia (Gaia Collaboration et al., 2016). Cette population est mal connue en raison des taux élevés de contamination et d'incomplétude dans ce régime d'observation. COSMIC-DANCE rassemble des données d'imagerie grand champ d'amas ouverts proches et de régions de formation d'étoiles à partir d'une large gamme d'observations au sol et d'archives de données. Ces données sont utilisées pour compiler des catalogues d'étoiles incluant des mesures de mouvement propres, c'est-à-dire le mouvement apparent des étoiles dans le ciel, et les probabilités d'appartenance à l'amas, c'est-à-dire la probabilité qu'une étoile appartienne à l'amas en question. Il est donc essentiel pour COSMIC-DANCE de disposer d'un outil de détection de source universel, capable de gérer la grande hétérogénéité des données à traiter. Des outils robustes et fiables sont également nécessaires pour gérer la qualité d'image inconnue et variable des données extraites des archives.

La mission Euclid utilisera un télescope spatial développé et exploité par l'ESA, embarquant des caméras grand champ optiques et PIR. Euclid vise principalement à comprendre la nature de la matière noire et de l'énergie noire en mesurant précisément l'expansion accélérée de l'Univers.

La mesure du cisaillement gravitationnel en régime faible des galaxies (la faible distorsion des formes des galaxies due à la déviation des rayons lumineux par des structures massives le long de la ligne de visée) et l'étude de la répartition des galaxies représentent deux sondes cosmologiques majeures qui seront utilisées par Euclid pour étudier la matière noire et l'énergie noire. Par conséquent, la détection robuste des galaxies et l'estimation précise de leurs positions et formes font partie des exigences principales de la mission (Amiaux et al., 2010).

En plus de ces relevés, de nombreux autres futurs relevés prévoient de collecter d'énormes quantités de données, rendant nécessaire la conception d'outils de détection de sources fiables, robustes, automatiques et rapides.

En pratique, on peut distinguer deux types de sources : les sources ponctuelles, c'est-à-dire les étoiles et les quasars, et les sources étendues, qui sont principalement des galaxies, mais qui peuvent aussi être des nébuleuses compactes ou des amas stellaires dont les étoiles ne sont pas résolues. Actuellement, il existe des méthodes optimales pour détecter des sources ponctuelles isolées, tels que les algorithmes basés sur le filtrage adapté (Woodward, 1953, 2014; Bertin and Arnouts, 1996). Cependant, l'efficacité de ces méthodes est très limitée dans d'autres régimes, comme dans les champs encombrés (c'est-à-dire lorsque la densité de la source est si élevée que les images de sources se recouvrent, un phénomène connu sous le nom de recouvrement, ou *blending*), ou lorsque les images sont contaminées par des défauts optiques, électroniques et environnementaux. Ces limitations sont illustrées sur la Figure. C.1.

L'encombrement stellaire est particulièrement problématique dans les champs de basse latitude galactique, où le bruit de confusion définit la limite de détection et domine largement les erreurs photométriques et astrométriques. La situation est encore plus grave dans le domaine PIR, où l'extinction due à la poussière interstellaire est considérablement réduite. Actuellement, les méthodes de détection de sources les plus performantes dans les images encombrées sont encore largement empiriques et consistent à soustraire de manière itérative les sources ponctuelles, des plus brillantes aux plus faibles, en utilisant un modèle de fonction d'étalement de point (ou *PSF* pour *point spread function*) (Stetson, 1987; Schechter et al., 1993; Zhang and Kainulainen, 2019). La détection des étoiles les plus faibles est également compliquée par la présence de contaminants. Parmi les contaminants les plus gênants, on peut compter les halos optiques, en particulier dans les caméras à grand champ; les rayons cosmiques et pixels chauds dans les images sous-échantillonnées; et les nébuleuses. Avec les algorithmes actuels, les plus petites zones contaminées peuvent être interpolées (Popowicz et al., 2013), à condition qu'elles aient été préalablement identifiées. Les contaminants les plus étendus comme les nébuleuses sont quant à eux traités par des techniques d'estimation de fond de ciel complexes (Popowicz and Smolka, 2015) ou des modèles bayésiens Knollmüller et al. (2018). Dans ce contexte, disposer d'outils fiables et versatiles est essentiel pour identifier les contaminants.

Contrairement aux étoiles qui sont des sources ponctuelles, les galaxies sont des sources étendues. Pour les objets étendus, la complétude ne dépend pas seulement de la magnitude des sources, c'est-à-dire de leur flux total, mais également de leur brillance de surface, c'est-à-dire de la mesure de la luminosité par unité de surface du détecteur (ou angle solide). La fonction de sélection de détection des galaxies est donc bidimensionnelle (Driver et al., 2005). Elle est illustrée en Figure. C.2. Même lorsqu'elles sont isolées, les galaxies à faible brillance de surface peuvent donc facilement être manquées par de simples algorithmes de seuillage fonctionnant à une seule échelle de détection, comme le montre la Figure. C.2.

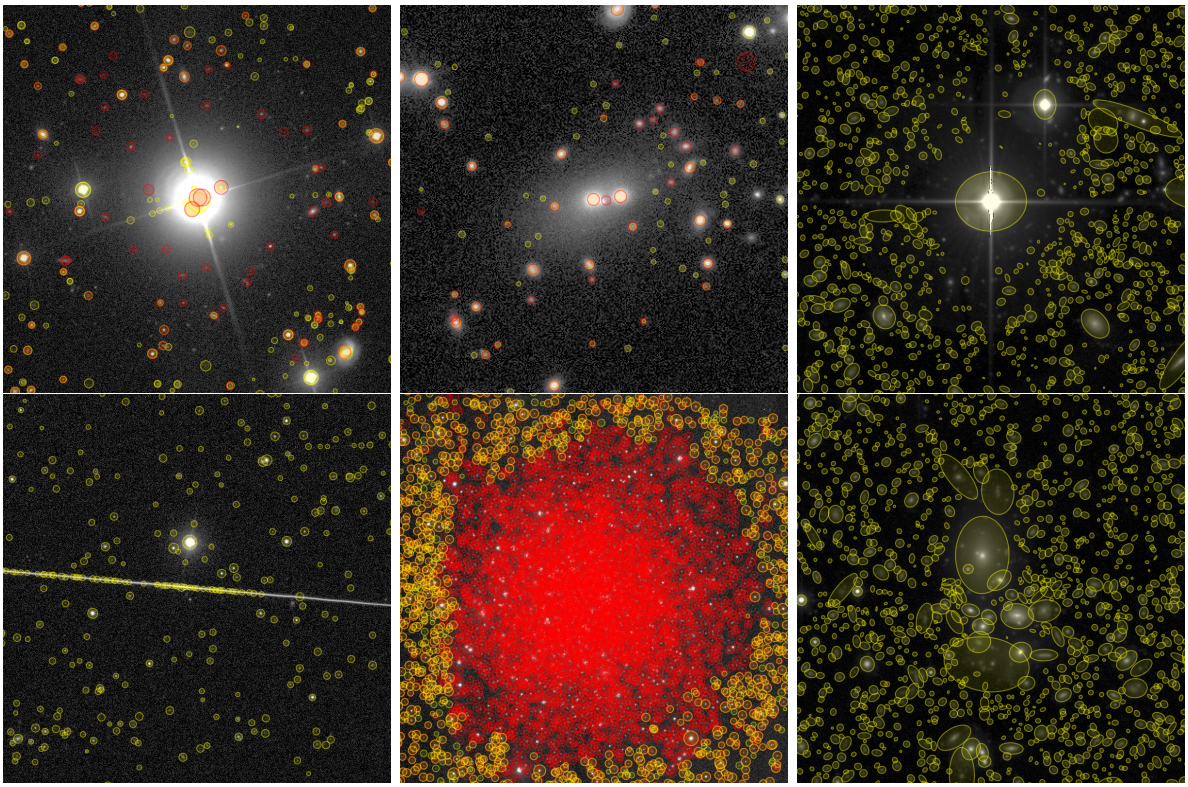


Figure C.1: Illustrations des principales limites des algorithmes de détection de sources actuels. Les cercles jaunes représentent les détections du SDSS (12th data release, Alam et al., 2015) tandis que les cercles rouges représentent les détections de Pan-STARRS (1st data release, Flewelling, 2017, 2018), à l'exception des images de droite qui présentent des détections de SExtractor dans des images du CFHTLS (Cuillandre and Bertin, 2006). Les images de gauche montrent des problèmes liès aux contaminants. En haut à gauche : exemples de fausses détections sur les aigrettes de diffraction des étoiles et dans le noyau saturé. On peut également noter que les sources autour de l'étoile brillante ne sont pas détectées. En bas à gauche : exemples de fausses détections sur une traînée traversant l'image. Les images au milieu montrent des problèmes de séparation de sources. En haut au milieu : la source voisine de la source centrale n'est pas bien détectée dans Pan-STARRS et même ratée dans le SDSS. En bas au milieu : le séparateur de sources de Pan-STARRS produit énormément de détections dans l'amas globulaire NGC 5466, alors que le SDSS ignore simplement cette zone. Les images de droite montrent à la fois des problèmes liés aux contaminants et à la séparation de sources. En haut à droite : les sources autour de l'étoile brillante sont ignorées. En bas à droite : les sources autour des sources plus étendues ne sont pas détectées. Les images sont vues via l'outil de visualisation Visiomatic 2 (Bertin et al., 2019a).

Pourtant, ces objets sont d'une grande importance en astrophysique. En effet, d'après la cosmologie observationnelle, les scénarios de formation de galaxies les plus probables dérivés du modèle de matière noire froide prédisent que ces objets sont abondants, à la fois sous forme de galaxies satellites ou dans les filaments perlés de galaxies, tous deux dominés par la matière noire (Kauffmann et al., 1993; Moore et al., 1999). Mise à part leur faible brillance de surface, la détection de tels objets est également compliquée par la présence de lumière intra-amas (Contini et al., 2014), de résidus de collision de galaxies tels que des coquilles (Hendel and Johnston, 2015) et des cirrus galactiques diffus provenant de nuages de poussière froide (Miville-Deschênes et al., 2016). Des illustrations de ces phénomènes sont présentées dans la Figure. C.3.

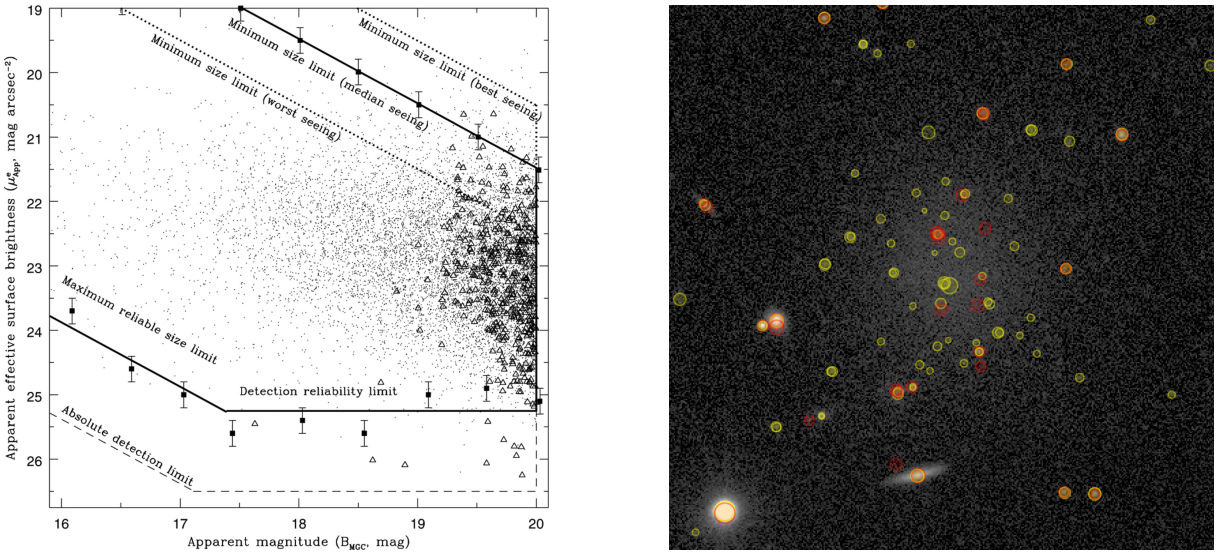


Figure C.2: À gauche : version simplifiée de la fonction de sélection de détection de galaxies dans le cadre du catalogue de galaxies Millennium (Driver et al., 2005). Étant donné que les galaxies sont des sources étendues, il existe une limite de détection en magnitude et une limite de détection en brillance de surface. La ligne inclinée en bas à gauche définit une limite de détection en taille maximale : si une galaxie est trop étendue, elle ne pourra pas être détectée. Les lignes inclinées en haut à droite définissent des limites de détection en taille minimale : si une galaxie est trop petite, elle peut être confondue avec une source ponctuelle, en particulier quand la qualité d'image (*seeing*) est mauvaise. À droite : un exemple de galaxie à faible brillance de surface non détectée dans les catalogues SDSS et Pan-STARRS. On peut aussi noter les fausses détections sur les pics de bruit dans la galaxie. Cette image est vue via Visiomatic 2 (Bertin et al., 2019a).

De nombreux efforts sont consacrés à l'inventaire et aux mesures de ces objets à travers des relevés en cours ou à venir, comme Dragonfly (Abraham and van Dokkum, 2014), Messier (Valls-Gabaud and MESSIER Collaboration, 2017), Huntsman (Spitler et al., 2019), MATLAS (Duc, 2020) et CASTLE (Lombardo et al., 2020). Il est aussi envisageable d'augmenter la détectabilité de ces objets en utilisant des approches multi-échelles (Starck et al., 2000). Cependant, outre les complications mentionnées ci-dessus, de nombreux contaminants de faible brillance de surface interfèrent avec la détection de ces galaxies dans la pratique, comme les halos d'étoiles, ainsi que les résidus de franges et de calibrations. Même à ce jour, comme il n'existe aucun algorithme automatique capable de fonctionner dans ce régime, l'inspection visuelle reste nécessaire (Bílek et al., 2020). Il est donc essentiel de développer des algorithmes de détection multi-échelles suffisamment "intelligents" pour gérer ces situations complexes.

De plus, l'encombrement affecte également la détection et les mesures des galaxies. Les galaxies ne sont pas distribuées indépendamment dans le ciel. Par l'action de la gravité sur les fluctuations de densité de l'Univers primordial, elles se répartissent en amas, feuilles et filaments. Les images de galaxies sont donc susceptibles de se recouvrir, voire de se mélanger, éventuellement avec des étoiles du premier plan. Ceci affecte fortement les statistiques dérivées des catalogues de galaxies, en particulier en cosmologie observationnelle, par exemple, lors des mesures de la fonction de corrélation des galaxies, de la richesse des amas (Gruen et al., 2019) ou des magnifications gravitationnelles (Gaztanaga et al., 2020). C'est ainsi qu'environ 20% des sources identifiées comme des galaxies dans les catalogues de relevés au sol les plus profonds finissent par être supprimées des ensembles de données de mesures des lentille gravitationnelles faibles à cause

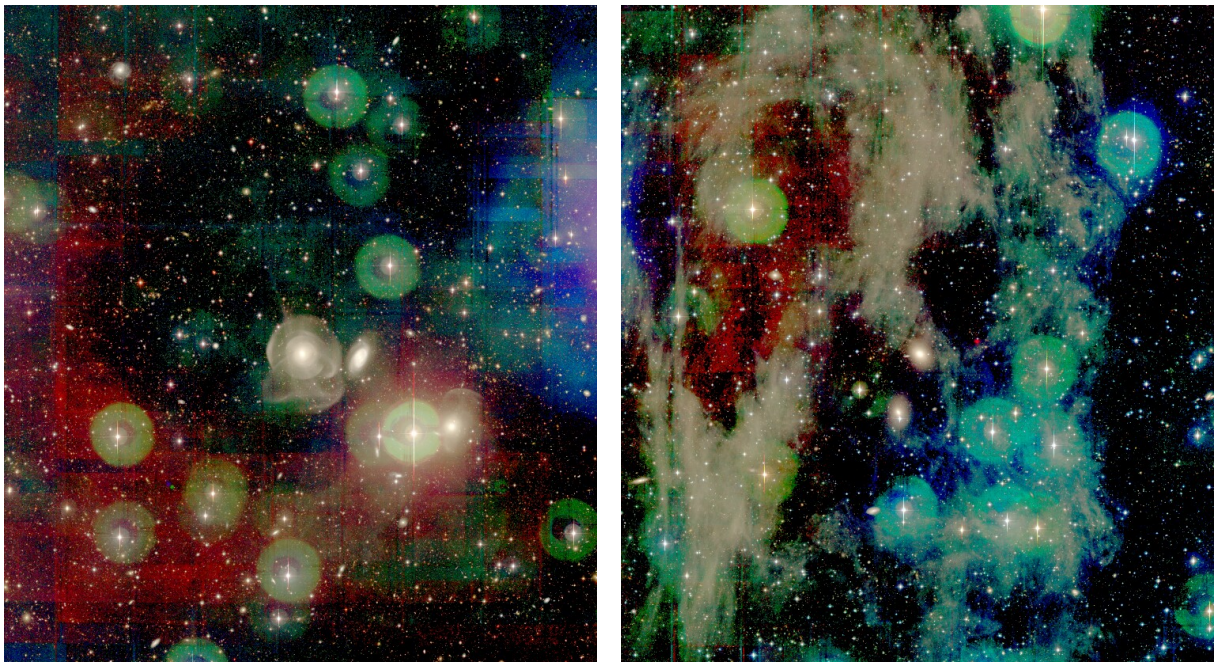


Figure C.3: Exemples d'images de galaxies du relevé MATLAS (Duc et al., 2015) illustrant des artefacts de faible brillance de surface. À gauche : NGC 0474. La galaxie montre plusieurs coquilles et flux radiaux. À droite : NGC 2592. Le cirrus est répandu sur la majeure partie du champ. D'autres images sont disponibles sur http://irfu.cea.fr/Projets/matlas/public/Atlas3D/atlas3d-pXXIX.html. Les images présentées ici sont vues via Visiomatic 2 (Bertin et al., 2019a).

du mélange des sources (Chang et al., 2013), même si cela concerne principalement les mesures de sources et non les détections. À ce niveau, les biais statistiques causés par les recouvrements de sources peuvent être estimés à l'aide de simulations d'images (Chang et al., 2015; Suchyta et al., 2016). Il est même envisageable de se libérer de ces biais grâce aux méthodes de calcul bayésiens approximatifs (Carassou et al., 2017; Kacprzak et al., 2020; Tortorelli et al., 2020).

Tous ces défis et contraintes vont devenir encore plus importants pour les prochain relevés à grande échelle comme Euclid ou le LSST. La quantité de données sans précédent que ces relevés vont fournir nécessite de nouveaux outils de détection de sources et de contaminants rapides, fiables et automatique.

Dans ces travaux de thèse, nous proposons d'aborder ces problèmes avec les approches orientées données qui ont émergées récemment. Plus particulièrement, nous aspirons à tirer parti des techniques d'apprentissage supervisé et des réseaux de neurones à convolutions (LeCun et al., 1995), dont les performances sont avérées dans les tâches de vision par ordinateur telles que la classification d'images (attribution d'étiquettes aux images, Krizhevsky et al., 2012; Simonyan and Zisserman, 2014), la segmentation d'images (attribution d'étiquettes aux pixels, Ronneberger et al., 2015; Badrinarayanan et al., 2017) et la détection d'objets INSTANCE AWARE (Redmon et al., 2016; Ren et al., 2015; He et al., 2017), où chaque objet est détecté individuellement et éventuellement segmenté. Il s'agit d'un changement de paradigme complet par rapport aux approches algorithmiques plus traditionnelles, modifiant la façon dont les problèmes sont abordés.

Ce manuscrit est divisé en chapitres organisés de la façon suivante : dans le Chapitre 2, je présente notre modèle décrivant les images grand champ optique et PIR. Cela nous aidera à identifier les caractéristiques principales des images astronomiques et posera le problème de la détection de sources. Après avoir examiné les solutions possibles à ce problème dans le Chapitre 3, je justifie notre choix d'approche basée sur l'apprentissage automatique. Dans le Chapitre

4, j'introduis les concepts nécessaires liés aux techniques d'apprentissage automatique supervisé que nous appliquons aux images : les réseaux de neurones à convolutions. Cela nous amènera au Chapitre 5, où j'aborde l'identification des contaminants avec MaxiMask et MaxiTrack. Dans le Chapitre 6 je me concentre le problème de détection de sources et présente notre nouveau prototype de détecteur basé sur des réseaux de neurones à convolutions. Enfin, je résume nos résultats et discute des futures lignes directrices des travaux dans le Chapitre 7.

Appendix D

Résumé substantiel

Modèle d'image : Nous commençons par expliciter le modèle d'image qui va nous servir à plusieurs égards. Ce modèle nous permet d'abord de comprendre comment les images astronomiques se forment, ainsi que de définir leurs caractéristiques pour poser le problème de détection de sources. Il est également important de connaître la formation des images astronomiques dans le but de pouvoir en simuler de manière réaliste.

Dans l'optique de concevoir un détecteur de sources universel, nous développons un modèle simplifié mais générique de la formation des images astronomiques. En faisant les hypothèses que le processus de formation d'image est linéaire et équivariant par translation et que tous les pixels du détecteurs sont indépendants, de même sensibilité et arrangés sur une grille homogène, nous avons :

$$\boldsymbol{y} = N(\text{Ш}_S(\boldsymbol{h} * \boldsymbol{x})) + \boldsymbol{n}, \tag{D.1}$$

où $\boldsymbol{y}$ est le signal observé, $*$ désigne l'opérateur de convolution, $\boldsymbol{h}$ est la réponse impulsionnelle de l'instrument ou fonction d'étalement de point, incluant les caractéristiques optiques de l'instrument et les perturbations de l'atmosphère dans le cas d'observations au sol, $\boldsymbol{x}$ est le vrai signal, $\text{Ш}_S()$ désigne la distribution cha d'échantillonnage (ou peigne de Dirac) de période S correspondant à la taille de pixel du détecteur, $N()$ désigne le bruit intrinsèque de Poisson lié au compte de photons et $\boldsymbol{n}$ est le bruit additionnel Gaussien de lecture du détecteur.

Les deux principaux types de sources à détecter dans les images sont les étoiles et les galaxies. Là où les premières sont ponctuelles et apparaissent comme une tache correspondant à la réponse impulsionnelle de l'instrument, les secondes sont étendues et peuvent présenter des structures complexes (bulbes, disques, bras spiraux) à plusieurs échelles.

Méthodes de détection de sources existantes : Dans le cas où les sources sont faibles par rapport au fond de ciel (ce qui est généralement le cas dans notre cadre d'observation d'images grand champ à longs temps d'exposition dans les domaines optique et proche infrarouge), nous pouvons faire l'approximation que le bruit des images est additif et stationnaire. Sous cette hypothèse, le filtre linéaire (corrélateur) qui maximise le rapport signal à bruit des sources présentes dans les images de manière optimale est le filtre adapté (Woodward, 1953, 2014; Turin, 1960). Celui-ci consiste à corréler l'image par le profil des sources recherchées, c'est-à-dire par la réponse impulsionnelle de l'instrument dans le cas de sources ponctuelles. Appliqué après d'éventuels prétraitements (consistant généralement à estimer et soustraire la composante de fond de ciel) et combiné à des méthodes de seuillage ou de détection de pics, c'est une des méthodes les plus utilisées pour la détection de sources (Bertin and Arnouts, 1996). Cependant, le filtrage adapté n'est optimal que pour détecter des sources ponctuelles isolées en présence d'un bruit stationnaire. Ainsi, dès lors que les champs sont encombrés, contaminés, ou présentent des objets à plusieurs échelles, le filtrage adapté devient limité.

Pour résoudre les problèmes liés à l'encombrement, des procédures de séparation de sources sont utilisées après la détection des sources (Bertin and Arnouts, 1996). Cependant, tout comme les méthodes déjà mentionnées, elles restent très empiriques et se basent sur des heuristiques, c'est-à-dire qu'elles ne sont pas garanties d'être optimales et que leurs paramètres doivent être adaptés à chaque application.

Pour pallier aux problèmes d'échelles, des approches multi-échelle principalement basées sur les ondelettes ont été développées Bijaoui and Rué (1995). Même si quelques-unes ont été utilisées en pratique avec des objectifs bien précis (Cayón et al., 2000; Starck et al., 2003), la fusion des composantes des sources identifiées à chaque échelle n'a jamais trouvé de réelle solution, limitant fortement ces approches. Cependant, leur développement a poussé des recherches vers les domaines d'acquisition comprimée et de codage parcimonieux (Bobin et al., 2008; Beckouche et al., 2013), dans lesquels on cherche à exprimer un signal à l'aide d'un dictionnaire de fonctions permettant de représenter des caractéristiques des images pertinentes pour effectuer un traitement donné.

En ce sens, nous suivons cet esprit en explorant des méthodes d'apprentissage supervisé, en particulier d'apprentissage profond, dont l'essence est d'apprendre une représentation des données pour résoudre une tâche. Nous nous dirigeons aussi vers ces méthodes dans l'espoir de concevoir des outils de détection de sources plus universels (pouvant s'adapter à divers instruments et conditions ambiantes sans avoir à faire des nombreux ajustements), plus robustes vis-à-vis des nombreuses complications au problème de détection de sources (contaminants, recouvrement de sources, caractère multi-échelle des objets) et de manière générale plus efficaces. En effet, ces méthodes ont largement prouvé leur potentiel dans de multiples domaines de vision par ordinateur, comme la classification d'images (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014) ou la segmentation d'images (Ronneberger et al., 2015; Badrinarayanan et al., 2017). Ces approches sont néanmoins très différentes des approches algorithmiques plus classiques. Ici, il s'agit d'apprendre une représentation des données à partir des données (brutes) elles-mêmes.

Apprentissage supervisé et réseaux de neurones à convolutions : Ayant à disposition une base de données contenant des couples entrée-sortie, le but de l'apprentissage automatique supervisé est d'apprendre à prédire les sorties à partir des entrées via un modèle. Le processus d'apprentissage consiste à itérer des étapes d'apprentissage. Le modèle est préalablement initialisé avec des paramètres aléatoires. À chaque étape d'apprentissage, une prédiction est effectuée et l'erreur de prédiction est quantifiée grâce à la connaissance de la vérité terrain et à une fonction de coût. En fonction de l'erreur de prédiction, tous les paramètres du modèle sont alors mis à jour via une méthode d'optimisation de manière à améliorer les futures prédictions.

Dans le cadre de ces travaux, nous utilisons comme modèles les réseaux de neurones multi-couches à propagation avant. Il s'agit de réseaux de neurones acycliques utilisant comme élément de base le neurone artificiel (McCulloch and Pitts, 1943). Un neurone artificiel est un noeud qui reçoit un certain nombre d'entrées, les multiplie par ses poids, ajoute un biais et applique une fonction d'activation. Ceux-ci sont organisés en couches : tous les neurones d'une couche donnée prennent comme entrée les sorties des neurones de la couche précédente, et envoient leur sortie aux neurones de la couche suivante. Les neurones d'une même couche ne sont pas connectés entre-eux. Les neurones de la couche de sortie produisent un vecteur dont les valeurs correspondent à des valeurs réelles pour résoudre une tâche de régression ou à des scores d'appartenance à des classes (assimilables à des probabilités d'appartenance selon les fonctions d'activation) pour résoudre une tâche de classification. Les poids et le biais de chaque neurone sont les paramètres qui sont appris par le réseau de neurones. La méthode de descente de gradient et l'algorithme de rétropropagation du gradient (Rumelhart et al., 1985, 1988) sont utilisés pour les mettre à jour à chaque étape d'apprentissage en fonction de l'erreur de prédiction calculée par la fonction de

coût.

Les réseaux de neurones multi-couches à propagation avant ne peuvent cependant pas être utilisés directement sur des images brutes dont la dimension (nombre de pixels) est beaucoup trop grande. Ce n'est qu'à l'arrivée des réseaux de neurones à convolution (LeCun et al., 1995), utilisant des couches convolutives et des couches de mise en commun, que cela a été possible. L'architecture classique d'un réseau de neurones à convolutions utilise un empilement de couches convolutives suivies de couches de mise en commun. L'opération de convolution, dont les valeurs de noyau sont les paramètres d'apprentissage, permet de traiter une image avec un petit nombre de paramètres et de détecter des caractéristiques particulières dans les images. Une fois convoluées et activées, les différentes versions de l'image d'entrée sont appelées cartes de caractéristiques. Comme de nombreuses convolutions de chaque carte de caractéristiques sont faites à chaque couche, des couches de mise en commun permettent de réduire leur taille au fil de l'avancée dans le réseau. La mise en commun permet également d'effectuer une sélection parmi les motifs détectés. Au fil des couches, le réseau combine les motifs qu'il détecte pour créer des représentations des données d'entrées de plus en plus complexes. Celle-ci sont finalement transmises à des couches complètement connectées, similaires à des réseaux de neurones multi-couches à propagation avant plus classiques, qui produisent la sortie du réseau.

Identification des contaminants : Dans l'optique de résoudre le problème de détection de sources de manière robuste, nous nous attaquons d'abord aux contaminants qui polluent les images. Ces contaminants sont nombreux et d'origines variées. Nous les classons en deux catégories principales. D'une part, les contaminants locaux, qui affectent les images au niveau du pixel, et d'autre part, les contaminants globaux, qui affectent les images en entier. Parmi les contaminants locaux nous considérons les contaminants liés à l'électronique du détecteur (mauvais pixels, saturation, effets de persistence, interférences entre ports de lecture), ceux liés à l'optique du téléscope (franges, aigrettes de diffraction, halos d'étoiles, lumière diffuse) et ceux liés à l'environnement (rayons cosmiques, traînées, nébuleuses). D'autre part, les contaminants globaux, qui affectent les images dans leur entiereté, et parmi lesquels nous pouvons principalement compter les erreurs de guidage de téléscope et les erreurs de mise au point.

Il existe peu d'approches qui s'attaquent aux contaminants de manière générique dans la littérature. La plupart des méthodes existantes se résument en deux catégories. D'abord, il existe des méthodes qui se focalisent sur un contaminant en particulier, comme les rayons cosmiques (van Dokkum, 2001) ou les traînées (Bektešević and Vinković, 2017). Ensuite, il y a les chaines de traitement des plus grands relevés qui utilisent une connaissance précise de leur instrument pour identifier les principaux contaminants (Bosch et al., 2018; Morganson et al., 2018). Certains grands relevés se basent aussi sur des observations multi-époque pour identifier les contaminants transitoires, comme ce sera le cas du LSST (Bosch et al., 2019). Cependant, aucune de ces approches ne reprend nos objectifs d'universalité consistant à concevoir un outil unique pouvant gérer un maximum de contaminants sans nécessiter d'importants ajustements selon les cas d'applications.

Pour réaliser ces ambitions, nous avons développé MaxiMask et MaxiTrack, deux réseaux de neurones à convolutions (Paillassa et al., 2020). MaxiMask effectue de la segmentation sémantique : il associe des étiquettes aux pixels d'une image (pour ce faire, les réseaux de neurones à convolution classiques incluent des couches de sur-échantillonnage pour retrouver la résolution d'image intiale et effectuer des prédictions à l'échelle du pixel). Actuellement, MaxiMask peut détecter les rayons cosmiques, les mauvais pixels, les effets de persistence, les traînées, les franges, les nébuleuses, les pixels saturés et les aigrettes de diffractions. MaxiTrack effectue quant à lui de la classification d'images et peut détecter la présence d'erreurs de guidage de téléscope.

Pour entrainer MaxiMask et MaxiTrack, nous avons principalement utilisé des données du relevé COSMIC-DANCE (Bouy et al., 2013) provenant de nombreux instruments ainsi que des simulations faites avec SkyMaker (Bertin, 2009). Nous construisons des échantillons d'apprentissage nous-mêmes en ajoutant des contaminants dans des images (réelles) non contaminées. De cette manière, nous savons exactement quels pixels sont affectés par quels contaminants et pouvons construire des masques de vérité terrain pour chacun des contaminants. Les pixels saturés et aigrettes de diffractions nécessitent néanmoins d'être préalablement identifiés dans les images non contaminées.

Un des principaux problème rencontré pour l'apprentissage de MaxiMask est la forte disproportion de classe : pour chaque contaminant, il y a beaucoup plus de pixels étiquetés non contaminés que de pixels étiquetés contaminés. Le réseau a ainsi un très fort a priori statistique pour classer les pixels et tombe facilement dans la solution qui consiste à tous les classer comme non contaminés. Nous réduisons cet effet de manière empirique en appliquant un poids au coût de chaque pixel en fonction de la représentativité de ses classes d'appartenance dans l'ensemble des échantillons d'entrainement.

Une fois entrainés, nous vérifions que MaxiMask et MaxiTrack ne souffrent pas de surapprentissage Nous évaluons leur performance sur des données de test en utilisant des mesures appropriées à la forte disproportion de classe. MaxiMask et MaxiTrack affichent des performances satisfaisantes et nous montrons que MaxiMask n'est pas limité au régime des images d'apprentissages qui contiennent la majorité des contaminants. Nous montrons aussi que MaxiMask est compétitif avec l'état de l'art au niveau de la détection de rayons cosmiques (McCully and Tewes, 2019).

Nous décrivons également une méthode pour adapter les probabilités de sortie de MaxiMask et MaxiTrack à des nouveaux a prioris, c'est-à-dire à des nouvelles proportions de contaminants dans les données. Ceci est possible dans un cadre Bayésien, où il est démontrable que sous l'hypothèse d'un apprentissage *parfait*, les réseaux de neurones classifieurs produisent des probabilités a posteriori (Richard and Lippmann, 1991; Hampshire II and Pearlmutter, 1991; Rojas, 1996).

Enfin, nous évaluons qualitativement les performances de MaxiMask en l'appliquant à des nouvelles données. MaxiMask est notamment capable de détecter les satellites Starlink (DECam), de s'adapter à un régime d'images sous-échantillonnées (HST) et détecter les rayons cosmiques dans des simulations Euclid. Notons qu'aucune image de ces deux derniers instruments n'est utilisée dans l'ensemble d'apprentissage de MaxiMask. MaxiMask et MaxiTrack sont également disponibles sous forme de modules d'inférence python sur GitHub[1].

Détection de sources : Dans la deuxième partie des travaux, nous abordons la détection de sources. S'étant déjà attaqué aux contaminants, il reste deux principaux défis à relever concernant la détection de sources : la séparation de sources et l'aspect multi-échelle des objets.

En plus de cela, le détecteur de sources ne doit pas seulement segmenter les sources dans les images mais doit pouvoir détecter chaque source individuellement, une propriété appelée conscience d'instance (*instance aware*). Malgré l'abondance de méthodes développées en apprentissage automatique profond sur le sujet, nous ne trouvons pas d'approche satisfaisante pour une application directe à la détection de sources, en particulier en ce qui concerne la séparation de sources.

Nous concevons donc une nouvelle approche multi-échelle de détection de sources basée sur les réseaux de neurones à convolutions. Le principe de notre détecteur est d'identifier chaque source par une empreinte d'une seule composante assez petite pour la distinguer des autres sources par analyse en composante connectées (Rosenfeld and Pfaltz, 1966). Le réseau effectue

[1] https://github.com/mpaillassa/MaxiMask

de la segmentation sémantique : chaque source est identifiée par son empreinte dans un plan correspondant à son échelle. Le coeur de l'approche réside alors dans la définition des empreintes de sources et dans la procédure d'affectation des sources aux échelles. Après de nombreuses expérimentations, nous retenons comme définition d'empreinte de source les pixels plus brillants formant 25% du flux total de la source. L'affectation de chaque source à une des trois échelles gérées par le réseau se fait par octave en fonction des tailles d'empreinte des sources : les sources dont l'empreinte représente moins de 9 pixels sont affectées à la première échelle, celles dont l'empreinte représente entre 9 et 36 pixels sont affectées à la deuxième échelle, et les restantes sont affectées à la troisième.

Pour entrainer notre détecteur, nous construisons des échantillons d'apprentissage en simulant des images entières à partir d'images non bruitées de sources isolées. Pour les sources ponctuelles, nous utilisons le logiciel SkyMaker (Bertin, 2009). Dans le cas des sources étendues, nous utilisons des simulations de cônes de lumière provenant de simulations à N corps (C.Laigle, communication privée). Grâce à ces images non bruitées, nous pouvons à la fois calculer les empreintes et échelles de chaque source et simuler des images astronomiques réalistes.

Après entrainement, nous évaluons les performances de notre détecteur et les comparons à SExtractor. En mesurant les taux de complétude et de contamination à différents seuils de détection sur des échantillons de test, nous montrons que notre détecteur présente de meilleures performances que SExtractor, que ce soit dans un régime d'images contaminées ou pas. En particulier, la présence de contaminants dans les échantillons d'apprentissage de notre détecteur le rend naturellement robuste à ceux-ci, ce qui n'est pas le cas de SExtractor qui nécessite des outils externes pour gérer les contaminants. Malgré les limitations de nos échantillons d'apprentissage, nous notons également que le détecteur affiche des performances très satisfaisantes à l'occasion de tests sur des données réelles, laissant entrevoir de très bonnes perspectives pour cette approche.

Appendix E

Conclusion en français

Dans cette thèse, nous avons conçu de nouveaux algorithmes pour extraire des catalogues plus fiables à partir des images astronomiques. En tirant parti des techniques d'apprentissage profond et des réseaux de neurones à convolutions (LeCun et al., 1995; Krizhevsky et al., 2012; Badrinarayanan et al., 2017), nous avons développé des modèles de pointe qui peuvent être facilement appliqués à une large gamme d'images optiques et PIR grand champ de manière entièrement automatisée.

Ce travail a été axé en grande partie sur les données et illustre une nouvelle approche pour développer des outils de traitement de données. En effet, les nouvelles approches orientées données reposent principalement sur des ensembles de données et des procédures d'entrainement conçus avec minutie. Les ensembles de données doivent être suffisamment grands et représentatifs de la tâche à résoudre pour permettre la généralisation, c'est-à-dire la capacité des modèles entrainés à bien fonctionner sur de nouvelles données. Pendant la durée de la thèse, plus de 50 To de données d'images hétérogènes ont été traitées, à la fois par des programmes d'analyse de données personnalisés et par des procédures d'apprentissage.

Nos premières réalisations, MaxiMask et MaxiTrack (Paillassa et al., 2020), sont des détecteurs de contaminants. MaxiMask et MaxiTrack permettent d'identifier une grande variété de défauts dans les images, et peuvent être utilisés en complément des algorithmes de détection de sources traditionnels ou pour un contrôle automatisé de qualité d'image. Pour entrainer les réseaux MaxiMask et MaxiTrack, nous avons construit des ensembles de données réalistes couvrant une grande diversité d'images en utilisant des données extraites de divers instruments du relevé COSMIC-DANCE, ainsi que des simulations d'images.

Pour faire face aux problèmes de déséquilibre de classe présents dans les image astronomiques, nous avons défini une stratégie empirique de pondération des coûts des pixels et proposé une méthode de réequilibrage des sorties du détecteur basé sur une approche Bayésienne. Cette approche permet de gérer différents régimes d'image en mettant simplement à jour un ensemble d'a prioris, et semble bien fonctionner dans la pratique.

Notre analyse des inférences de MaxiMask a montré qu'il généralise bien sur des données réelles, y compris pour des données provenant d'instruments non utilisés pour l'entraînement. MaxiMask et MaxiTrack sont publiquement accessibles en tant que modules d'inférence[1].

Nous sommes conscients que des contaminants particulièrement problématiques pour les images grand champ, tels que les halos optiques, les reflets et lumières diffuses ne sont pas encore pris en compte dans MaxiMask. Comme ceux-ci sont difficiles à simuler ou à isoler à partir d'images, la construction d'un ensemble d'apprentissage nécessite une quantité de travail importante, similaire à ce qui a été fait pour les aigrettes de diffraction. C'est un projet que j'envisage d'entreprendre dans un futur proche, car je travaillerai sur des images HSC dans le cadre de mes

[1] https://github.com/mpaillassa/MaxiMask

recherches post-doctorales. À condition que des données supplémentaires soient disponibles et incorporées dans l'ensemble d'apprentissage, d'autres contaminants problématiques tels que les interférences entre ports de lecture pourraient également être pris en compte par MaxiMask, tandis que les performances sur des contaminants déjà pris en compte pourraient être améliorées, en particulier pour les aigrettes de diffraction.

Notre deuxième réalisation est un détecteur de sources astronomiques robuste et multi-échelle. Notre détecteur est capable de traiter divers problèmes qui entravent les algorithmes de détection de sources traditionnels tels que les objets étendus et à faible luminosité de surface, le recouvrement de sources et la contamination par des défauts. L'aspect multi-échelle de notre approche permet au détecteur d'identifier des sources à différentes échelles dans des cartes de sortie différentes. Nous avons évalué les performances du détecteur sur des données de test et avons trouvé une complétude 30 à 100% plus grande, ainsi qu'un taux de contamination environ 20 fois plus faible par rapport à un algorithme d'extraction de source traditionnel (SExtractor). Un article est en préparation et le détecteur de source sera disponible pour inférence sur GitHub.

Des travaux sont également en cours pour l'implémenter dans le logiciel SourceXtractor++ (Bertin et al., 2019b), qui sera largement utilisé pour analyser les données d'imagerie provenant de diverses sources dans le cadre de la mission Euclid. Comme les cartes d'empreintes de sortie du détecteur sont très similaires aux cartes de segmentation produites par SourceXtractor++, elles peuvent pratiquement les remplacer directement. De plus, elles fournissent des informations additionnelles pour les procédures de mesures de sources.

Bien que les performances actuelles du détecteur soient déjà satisfaisantes, des améliorations sont possibles au niveau des données. En particulier, nous sommes conscients que nos images de galaxies et leur distribution sont irréalistes, ce qui impacte nécessairement le comportement du détecteur. Des travaux supplémentaires sont clairement nécessaires pour inclure des modèles astrophysiques dans les simulations. De plus, et bien que ce ne soit pas notre objectif principal, des ensembles d'entrainement spécifiques pourraient également être construits pour cibler des instruments ou des objectifs scientifiques en particulier.

Enfin, il est important de noter que les catalogues produits par un détecteur multi-échelles, robuste et adaptatif comme le nôtre ont potentiellement une fonction de sélection plus complexe que, par exemple, des algorithmes plus simples basés sur des seuils en brillance de surface ou magnitude. Des ensembles de données d'images simulées seront donc également nécessaires pour dériver les fonctions de sélection et tirer pleinement parti des catalogues.

Appendix F

MaxiMask and MaxiTrack: Two new tools for identifying contaminants in astronomical images using convolutional neural networks

A&A 634, A48 (2020)
https://doi.org/10.1051/0004-6361/201936345

Astronomy & Astrophysics

MaxiMask and MaxiTrack: Two new tools for identifying contaminants in astronomical images using convolutional neural networks

M. Paillassa[1], E. Bertin[2], and H. Bouy[1]

[1] Laboratoire d'astrophysique de Bordeaux, Univ. Bordeaux, CNRS, B18N, allée Geoffrey Saint-Hilaire, 33615 Pessac, France
e-mail: maxime.paillassa@u-bordeaux.fr
[2] Sorbonne Université, CNRS, UMR 7095, Institut d'Astrophysique de Paris, 98 bis bd Arago, 75014 Paris, France

Received 18 July 2019 / Accepted 9 December 2019

ABSTRACT

In this work, we propose two convolutional neural network classifiers for detecting contaminants in astronomical images. Once trained, our classifiers are able to identify various contaminants, such as cosmic rays, hot and bad pixels, persistence effects, satellite or plane trails, residual fringe patterns, nebulous features, saturated pixels, diffraction spikes, and tracking errors in images. They encompass a broad range of ambient conditions, such as seeing, image sampling, detector type, optics, and stellar density. The first classifier, MaxiMask, performs semantic segmentation and generates bad pixel maps for each contaminant, based on the probability that each pixel belongs to a given contaminant class. The second classifier, MaxiTrack, classifies entire images and mosaics, by computing the probability for the focal plane to be affected by tracking errors. We gathered training and testing data from real data originating from various modern charged-coupled devices and near-infrared cameras, that are augmented with image simulations. We quantified the performance of both classifiers and show that MaxiMask achieves state-of-the-art performance for the identification of cosmic ray hits. Thanks to a built-in Bayesian update mechanism, both classifiers can be tuned to meet specific science goals in various observational contexts.

Key words. methods: data analysis – techniques: image processing – surveys

1. Introduction

Catalogs extracted from astronomical images are at the heart of modern observational astrophysics. Minimizing the number of spurious detections in these catalogs has become increasingly important because the noise added by such contaminants can, in many cases, compromise the scientific objectives of a survey. Properly identifying and flagging spurious detections yields substantial scientific gains, but it is complicated by the numerous types of contaminants that pollute images. Some of them stem from the detector electronics (e.g., dead or hot pixels, persistence, saturation), from the optics (diffraction along the optical path, scattered and stray light), from post-processing (e.g., residual fringes), while others are the results of external events (cosmic rays, satellites, tracking errors). The amount of data produced by modern astronomical surveys makes visual inspection impossible in most cases. For this reason, developing fully automated methods to separate contaminants from true astrophysical sources is a critical issue in modern astronomical survey pipelines.

Most current pipelines rely on a fine prior knowledge of their instruments to detect and mask electronic contaminants (e.g., Bosch et al. 2018; Morganson et al. 2018) and to some extent optical contaminants (e.g., Kawanomoto et al. 2016a,b). Cosmic ray hits can be identified by rejecting outliers in the timeline, provided that multiple consecutive exposures are available, by using algorithms sensitive to their peculiar shapes, such as Laplacian edge detection (e.g., LA Cosmic, van Dokkum 2001) or wavelets (e.g., Ordénovic et al. 2008). The Radon transform or the Hough transform have often been used to detect streaks caused by artificial satellites or planes in images (e.g, Vandame 2002; Nir et al. 2018).

In this work, we want to overcome some of the drawbacks of the above mentioned methods. First, the typical data volume produced by modern surveys requires that the software is largely unsupervised and as efficient as possible. Second, we aim to develop a robust and versatile tool for the community at large and therefore want to avoid the pitfall inherent in software that is tailored to a single or a handful of instruments, without compromising on performance. Third, we would like to have a unified tool able to detect many contaminants at once. Finally, we want to assign to each pixel a probability of belonging to a given contaminant class rather than Boolean flags. These constraints lead us to choose machine learning techniques and in particular supervised learning and convolutional neural networks (CNNs).

Supervised learning is a field of machine learning dealing with models that can learn regression or classification tasks based on a data set containing the inputs and the expected outputs. During the learning process, model parameters are adjusted iteratively to improve the predictions made from the input data. The learning procedure itself consists of minimizing a loss function that measures the discrepancy between model predictions and the expected values. Minimization is achieved through stochastic gradient descent. We recommend Ruder (2016) for an overview of gradient descent based optimization algorithms.

Convolutional neural networks (LeCun & Bengio 1995) are particulary well-suited for identifying patterns in images. Unlike previous approaches that would involve hand crafted feature detectors, such as SIFT descriptors (Lowe 1999), CNN models operate directly on pixel data. This is made possible by the use of trainable convolution kernels to detect features in images. Convolution is shift-equivariant, which allows the same features to be detectable at any image location.

A&A 634, A48 (2020)

CNNs are now widely used in various computer vision tasks, including image classification, that is assigning a label to a whole image (Krizhevsky et al. 2012; Simonyan & Zisserman 2014; Szegedy et al. 2015), and semantic segmentation, that is assigning a label to each pixel (Long et al. 2015; Badrinarayanan et al. 2017; Garcia-Garcia et al. 2017).

In this work, we propose to identify contaminants using both image classification and semantic segmentation.

In the following, we first describe the images that we used and how we built our data sets. Then, we focus on the neural network architecture that we used. Finally, we evaluate the models performance on test sets and on real data.

2. Data

In this section we describe the data used to train our two neural networks. We distinguish between two types of contaminants: On the one hand, local contaminants, that affects only a fraction of the image at specific locations. This includes cosmic rays, hot columns and lines, dead columns and lines, dead clustered pixels, hot pixels, dead pixels, persistence, satellite trails, residual fringe patterns, "nebulosity", saturated pixels, diffraction spikes, and over scanned pixels. These add up to 12 classes. On the other hand, global contaminants, that affects the whole image, such as tracking errors.

2.1. Local contaminant data

For local contaminants, we choose to build training samples by adding defects to uncontaminated images in order to have a ground truth for each contaminant. In this section we first describe the library of astronomical images used for our analysis, then focus on the selection of uncontaminated images, and finally describe the way each contaminant is added.

2.1.1. Library of real astronomical images

In an effort to have the most realistic dataset, we choose to use real data as much as possible and take advantage of the private archive of wide-field images gathered for the COSMIC-DANCE survey (Bouy et al. 2013). The COSMIC-DANCE library offers several advantages. First, it includes images from many past and present optical and near-infrared wide-field cameras. Images cover a broad range of detector types and ground-based observing sites, ensuring that our dataset is representative of most modern astronomical wide-field instruments. Table 1 gives an overview of the properties of the cameras used to build the image database. Second, most problematic exposures featuring tracking/guiding loss, defocusing or strong fringing were already identified by the COSMIC-DANCE pipeline, providing an invaluable sample of real problematic images.

In all cases except for Megacam, DECam, UKIRT and HSC exposures, the raw data and associated calibration frames were downloaded and processed using standard procedures with an updated version of *Alambic* (Vandame 2002), a software suite developed and optimized for the processing of large multi-chip imagers. In the case of Megacam, the exposures processed and calibrated with the *Elixir* pipeline were retrieved from the CADC archive (Magnier & Cuillandre 2004). In the case of DECam, the exposures processed with the community pipeline were retrieved from the NOAO public archive (Valdes et al. 2014). UKIRT exposures processed by the Cambridge Astronomical Survey Unit were retrieved from the WFCAM Science Archive. Finally, the HSC raw images were processed using the official HSC pipeline (Bosch et al. 2018). In all cases, a bad pixel map is associated to every individual image. In the case of DECam and HSC, a data quality mask is also associated to each individual image and provides integer-value codes for pixels which are not scientifically useful or suspect, including in particular bad pixels, saturated pixels, cosmic ray hits, satellite tracks, etc. All the images in the following consist of individual exposures and not co-added exposures.

Table 1. Instruments used in this study.

Telescope	Instrument	Type	Platescale [pixel^{-1}]	Ref.
CTIO Blanco	DECam	CCD	0.″26	(1)
CTIO Blanco	MOSAIC2	CCD	0.″26	(2)
KPNO Mayall	MOSAIC1	CCD	0.″26	(2)
KPNO Mayall	NEWFIRM	IR	0.″4	(3)
CFHT	MegaCam	CCD	0.″18	(4)
CFHT	CFH12K	CCD	0.″21	(5)
CFHT	UH8K	CCD	0.″21	(6)
INT	WFC	CCD	0.″33	(7)
UKIRT	WFCAM	IR	0.″4	(8)
LCO Swope	Direct CCD	CCD	0.″43	(9)
VST	OmegaCam	CCD	0.″21	(10)
Subaru	HSC	CCD	0.″17	(11)
VISTA	VIRCAM	IR	0.″34	(12)

References. (1) Flaugher et al. (2010); (2) Wolfe et al. (2000); (3) Autry et al. (2003); (4) Boulade et al. (2003); (5) Cuillandre et al. (2000); (6) Metzger et al. (1995); (7) Ives (1998); (8) Casali et al. (2007); (9) Rheault et al. (2014); (10) Kuijken et al. (2002); (11) Miyazaki et al. (2018); (12) Dalton et al. (2006).

2.1.2. Non-contaminated images

None of the exposures in our library are defect-free. The first step to create the non-contaminated dataset to be used as "reference" images consists in identifying the cleanest possible subset of exposures. CFHT-Megacam (u, r, i, z bands), CTIO-DECam (g, r, i, z, Y bands) and Subaru-HSC (g, r, i, z, y bands) exposures are found to have the best cosmetics and are selected to create the non-contaminated dataset. The defects inevitably present in these images are handled as follows.

First, dead pixels and columns are identified from flat-field images and inpainted using Gaussian interpolation (e.g., Williams et al. 1998). Then, the vast majority of cosmic rays are detected using the Astro-SCRAPPY Python implementation (McCully et al. 2018) of LA Cosmic (van Dokkum 2001) and also inpainted using Gaussian interpolation. Finally, given the high performance of the DECam and HSC pipelines, the corresponding images are perfect candidates for our non-contaminated datasets. These two pipelines not only efficiently detect but also interpolate problematic pixels (in particular saturated pixels, hot and bad pixels, cosmic ray hits). Such interpolations being a feature of several modern pipelines (e.g., various NOAO pipelines, but also the LSST pipeline), we choose to treat these pixels as regular pixels so that the networks are able to work with images originating from such pipelines.

Patches of size 400×400 pixels are randomly extracted from the cleaned images. 75% of them are used to generate training data and the remaining 25% for test data.

The final non-contaminated dataset includes 50 000 individual images, ensuring that we have a sufficiently diverse and large amount of training data for our experiment.

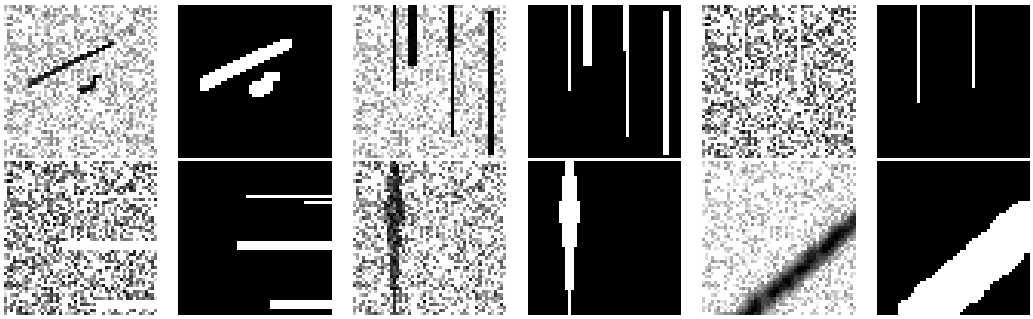


Fig. 1. Examples of contaminants and their ground truth. *Top row*: cosmic ray hits, hot columns, bad columns. *Bottom row*: bad lines, persistence, satellite trails.

A non-representative training set can severely impact the performance of a CNN and result in significant biases in the classification task. To prevent this, we measure a number of basic properties describing prototypical aspects of ground-based astronomical images to verify that their distributions in the uncontaminated dataset are wide enough and reasonably well sampled.

The measured properties include, for example, the average full-width at half-maximum (FWHM) of point-sources is estimated in each image using PSFEx (Bertin 2013). This allows us to ensure that the training set covers a broad range of ambient (seeing) conditions and point spread functions (PSFs) sampling. Also, the source density (number of sources in the image divided by the physical size of the image) is measured to make sure that our training set encompasses a broad range of source crowding, from sparse cosmological fields to dense, low-galactic latitude stellar fields.

Additionally, the background is modeled in all the images following the method used by SExtractor (Bertin & Arnouts 1996), i.e. using a combination of $\kappa.\sigma$-clipping and mode estimation. The background model provides important parameters such as the standard deviation of the background which is required in most of the data-processing operations that follow.

2.1.3. Cosmic rays (CR)

"Cosmic ray" hits are produced by particles hitting the detector or by the photons resulting from the decay of radioactive atoms near the detector. They appear as bright and sharp patterns with shapes ranging from dots affecting one or two pixels to long wandering tracks commonly referred to as "worm", depending on incidence angle and detector thickness.

We create a library of real CRs using dark frames with long exposure times from the CFH12K, HSC, MegaCam, MOSAIC, and OmegaCam cameras. These cameras comprise both "thick", red-sensitive, deep depletion charged-couple devices (CCDs), more prone to long worms, and thinner, blue-sensitive devices, more prone to unresolved hits. Dark frames are exposures taken with the shutter closed, so that the only contributors to the content of undamaged pixels are the offset, dark current, and CR hits (plus Poisson and readout noise). A mask $\boldsymbol{M}$ of the pixels affected by CR hits in a given dark frame $\boldsymbol{D}$ can therefore easily be generated by applying a simple detection threshold. We conservatively set this threshold to $3\sigma_{\rm D}$ above the median value $m_{\rm D}$ of D:

$$\forall p,\ M_{\rm p} = \begin{cases} 1 & \text{if } D_{\rm p} > m_{\rm D} + 3\sigma_{\rm D} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Among all the dark images used, a bit more than 900 million cosmic ray pixels are detected after thresholding. Considering that the average footprint area of a cosmic ray hit is 15 pixels, this represents a richly diversified population of about 60 million cosmic ray "objects".

Next we dilate $\boldsymbol{M}$ with a 3×3 pixel kernel to create the final $\boldsymbol{M}^{(\rm D)}$ mask. This mask is used both as ground truth for the classifier, and also to generate the final "contaminated" image $\boldsymbol{C}$ by adding CR pixels with rescaled values to the uncontaminated image $\boldsymbol{U}$:

$$\boldsymbol{C} = \boldsymbol{U} + k_{\boldsymbol{C}} \frac{\sigma_{\boldsymbol{U}}}{\sigma_{\boldsymbol{D}}} \boldsymbol{D} \odot \boldsymbol{M}^{(\boldsymbol{D})}, \quad (2)$$

where σ_U is the estimated standard deviation of the uncontaminated image background, $\odot$ denotes the element-wise product and k_C is a scaling factor empirically set to 1/8. $\boldsymbol{D}$ has been background-subtracted before this operation, using a SExtractor-like background estimation.

A typical CR hit added to an image and its ground truth mask are shown in Fig. 1.

2.1.4. Hot columns and lines, dead columns, lines, and clustered pixels, hot pixels, and dead pixels (HCL, DCL, HP, DP)

These contaminants mainly come from electronic defects and the way the detectors are read. They correspond to pixels having a response very different from that of neighbors, either much lower (bad pixels, traps) or much noisier (hot pixels). These blemishes can be found as single pixels, in small clusters, or affecting a large fraction of a column or row. We treat single pixels and clumps, columns, and lines separately, although they may often share a common origin.

All these hot or dead pixels added to the uncontaminated images are simulated. The number of these pixels is set as follow.

For columns and lines, a random number of columns and lines is chosen with a uniform distribution over [1,4]. Each column or line has a uniform length picked between 30 and the whole image height or width. It has a uniform thickness in [1,3]. For punctual pixels, a random fraction of pixels is chosen with a uniform distribution between 0.0002 and 0.0005. Pixels are uniformly distributed over the image. Clustered pixels are given a rectangular or a random convex polygonal shape. The random convex shapes are constrained to have 5 or 6 edges and to fit in 20×20 bounding boxes.

The values of these pixels are computed as follows. For hot values, a uniformly distributed random base value v is chosen in the interval $[15\sigma_U, 100\sigma_U]$. Then hot values are generated according to the normal law $\mathcal{N}(v, (0.02v)^2)$ so that hot values are randomly distributed over $[0.9v, 1.1v]$. For dead values, one of the following three equiprobable recipes is chosen at random to generate bad pixel values. Either all values are exactly 0. Either values are generated according to the normal law $\mathcal{N}\left(0, (0.02\sigma_U)^2\right)$ so that these are close to 0 values but not exactly 0. Either a random base value v is chosen with a uniform distribution in the interval $[0.1m_U, 0.7m_U]$, where m_U is the median of the uncontaminated image sky background. In this case, dead pixel values are generated using the normal law $\mathcal{N}\left(v, (0.02v)^2\right)$, so that values fall in the interval $[0.9v, 1.1v]$.

Example of such column and line defaults are shown in Fig. 1.

2.1.5. Persistence (P)

Persistence occurs when overly bright pixels in a previous exposure leave a remnant image in the following exposures.

To simulate this effect in an uncontaminated image, we applied the so-called "Fermi model" described in Long et al. (2015). Persistence, in units of $e^-.s^{-1}$), is modeled as a function of the initial pixel level x_p and time t:

$$f(x_\mathrm{p}, t) = A_\mathrm{p}\left(\frac{1}{\exp\left(-\frac{x_\mathrm{p}-x_0}{\delta x}\right)+1}\right)\left(\frac{x_\mathrm{p}}{x_0}\right)^{\alpha}\left(\frac{t}{1000}\right)^{-\gamma}. \tag{3}$$

The goal of Long et al. (2015) was to fit the model parameters $x_0, \delta x, \alpha, \gamma$ using observations to later predict persistence for their detector. In our simulations, parameter values are randomized to represent various types and amounts of persistence (see Table 2). To compute the pixel value of the persistence effect, we derive the number of electrons emitted by the persistence effect during the exposure. In the following, we note T the duration of the exposure in which the persistence effect occurs, and Δt the delay between that exposure and the previous one. We obtain the number of ADUs collected at pixel p during the interval $[\Delta t, \Delta t + T]$ by integrating Eq. (3) and dividing by the gain G:

$$P_\mathrm{p} = \frac{1}{G}\int_{\Delta t}^{\Delta t+T} f(x_\mathrm{p}, t)\,\mathrm{d}t \tag{4}$$

$$= \frac{A_\mathrm{p}}{G}\left(\frac{1000^{\gamma}}{\exp\left(-\frac{x_\mathrm{p}-x_0}{\delta x}\right)+1}\right)\left(\frac{x_\mathrm{p}}{x_0}\right)^{\alpha}\left(\frac{(\Delta t+T)^{1-\gamma}-\Delta t^{1-\gamma}}{1-\gamma}\right). \tag{5}$$

These pixel values are then added to the uncontaminated image:

$$\boldsymbol{C} = \boldsymbol{U} + k_{\boldsymbol{P}}\,\sigma_U\frac{\boldsymbol{P}-P_\mathrm{min}}{(P_\mathrm{max}-P_\mathrm{min})}, \tag{6}$$

where $\boldsymbol{P}$ are the persistence values computed in Eq. (5), P_min and P_max are the minimum and maximum of these values, and $k_{\boldsymbol{P}}$ is a scaling factor empirically set to 5.

Images of saturated stars are simulated using SkyMaker (Bertin 2009) and binarized to generate masks of saturated pixels. The masks define the footprints of persistence artifacts, within which the x_p's are computed (Table 2). An example is shown in Fig. 1.

Table 2. Parameters used for the generation of persistence.

A_p	1
x_p (e^-)	Poisson(x_m) with $x_\mathrm{m} \sim \mathcal{N}(15.10^5, (0.02 \times 15.10^5)^2)$
x_0 (e^-)	$\mathcal{N}(9.10^4, (0.02 \times 9.10^4)^2)$
δ_x (e^-)	$\mathcal{N}(18.10^3, (0.02 \times 18.10^3)^2)$
α	0.178
γ	1.078
G ($e^-.s^{-1}$)	$\mathcal{N}(10, 1)$

2.1.6. Trails (TRL)

Satellites or meteors, and even planes crossing the field of view generate long trails across the frame that are quasi-rectilinear. We simulate these motion-blurred artifacts by generating close star images with identical magnitudes along a linear path using once again SkyMaker. We also generate a second population of trails with magnitude changes to account for satellite "flares". A random, Gaussian-distributed component with a $\approx$1 pixel standard deviation is added to every stellar coordinate to simulate jittering from atmospheric turbulence, so that the stars are not aligned along a perfect straight line. For meteors, defocusing must be taken into account (Bektešević et al. 2018). The amount of defocusing θ, expressed as the apparent width of the pupil pattern in arc-seconds, is:

$$\theta = \frac{180}{\pi} \times 3600 \times \frac{D}{d}, \tag{7}$$

where D is the diameter of the primary mirror, and d the meteor distance, both in meters. D and d are randomly drawn from flat distributions in the intervals $[2, 8]$ and $[80\,000, 120\,000]$, respectively.

The ground truth mask is obtained by binarizing the satellite image at a small and arbitrary threshold above the simulated background. This mask is then dilated using a 7×7 pixel structuring element.

To avoid any visible truncation, we add the whole simulated satellite image multiplied by a dilated version $\boldsymbol{M}^{(\boldsymbol{S})}$ of the ground truth mask to the uncontaminated image:

$$\boldsymbol{C} = \boldsymbol{U} + k_{\boldsymbol{T}}\frac{\sigma_U}{\sigma_{\boldsymbol{T}}}\boldsymbol{T} \odot \boldsymbol{M}^{(\boldsymbol{T})}, \tag{8}$$

where $\sigma_{\boldsymbol{S}}$ is the standard deviation of the satellite image background, σ_U the standard deviation of the uncontaminated image background, and $k_{\boldsymbol{T}}$ is a scaling factor empirically set to 6. An example of a satellite trail is shown in Fig. 1.

2.1.7. Fringes (FR)

Fringes are thin-film interference patterns occurring in the detectors. The irregular shape of fringes is caused by thickness variations within the thin layers. To add fringing to images, we use real fringe maps produced at the pre-processing level by Alambic for all the optical CCD cameras of Table 1. These reconstructed fringe maps are often affected by white noise, which we mitigate by smoothed using a top-hat kernel with diameter 7 pixels. The fringe pattern $\boldsymbol{F}$ can affect large areas in an image but not necessarily all the image. To reproduce this effect, a random 3rd-degree 2D polynomial envelope $\boldsymbol{E}$ that covers the whole image is generated. The final fringe envelope $\boldsymbol{E}^{(\mathrm{F})}$

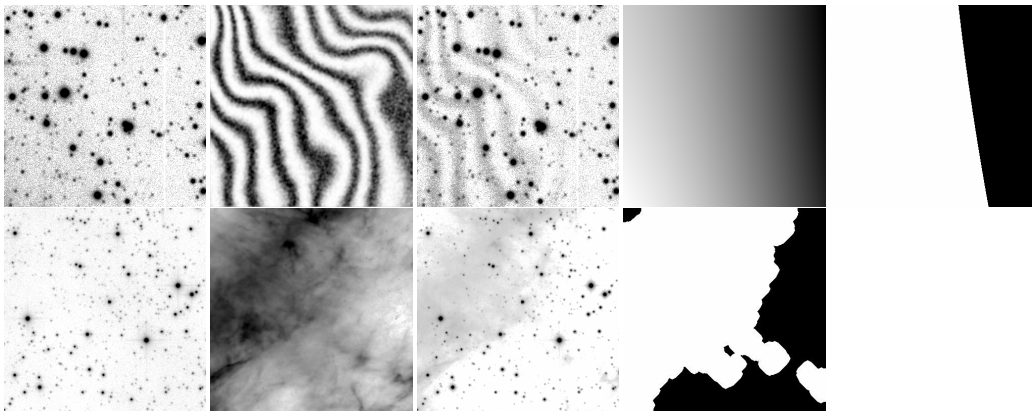


Fig. 2. Examples of added fringes and nebulosities. *Top*: fringes; uncontaminated input exposure, smoothed fringe pattern, contaminated image, ground truth mask, polynomial envelope. *Bottom*: nebulosities; uncontaminated input exposure, *Herschel* 250 μm molecular cloud image, contaminated image, ground truth mask.

is computed by normalizing $\boldsymbol{E}$ over the interval $[-5, 5]$ and flattening the result using the sigmoid function:

$$E_{\mathrm{p}}^{(\mathrm{F})} = \left(1 + \exp\left(-5\,\frac{2E_{\mathrm{p}} - E_{\max} - E_{\min}}{E_{\max} - E_{\min}}\right)\right)^{-1}, \tag{9}$$

where $E_{\min}$ and $E_{\max}$ are the minimum and maximum values of E_{p}, respectively.

The fringe pattern, modulated by its envelope, is then added to the uncontaminated image:

$$\boldsymbol{C} = \boldsymbol{U} + k_{\boldsymbol{F}}\frac{\sigma_{\boldsymbol{U}}}{\sigma_{\boldsymbol{F}}}\boldsymbol{F} \odot \boldsymbol{E}^{(\mathrm{F})}, \tag{10}$$

where $\sigma_{\boldsymbol{F}}$ is the standard deviation of the fringe pattern and $k_{\boldsymbol{F}}$ is an empirical scaling factor set to 0.6. The ground truth mask is computed by thresholding the 2D polynomial envelope to −0.20.

An example of a simulated contamination by a fringe pattern can be found in Fig. 2.

2.1.8. Nebulosity (NEB)

Extended emission originating from dust clouds illuminated by star light or photo-dissociation regions can be present in astronomical images. These "nebulosities" are not artifacts but they make the detection and measurement of overlapping stars or galaxies more difficult; they may also trigger the fringe detector. Hence, it is useful to have them identified and properly flagged. Because thermal distribution of dust closely matches that of reflection nebulae at shorter wavelength (e.g., Ienaka et al. 2013), we use far-infrared images of molecular clouds around star-forming regions as a source of nebulous contaminants. We choose pipeline-processed 250 μm images obtained with the SPIRE instrument (Griffin et al. 2010) on-board the *Herschel* Space Observatory (Pilbratt et al. 2010), which we retrieve from the *Herschel* Science Archive. The 250 μm channel offers the best compromise between signal-to-noise ratio and spatial resolution. Moreover, at wavelengths of 250 μm and above, low galactic latitude fields contain mostly extended emission from the cold gas and almost no point sources (apart from a few proto-stars and proto-stellar cores). Therefore, they are perfectly suited to being added to our optical and near-infrared wide-field exposures. We do not resize or reconvolve the SPIRE images, taking advantage of the scale-invariance of dust emission observed down to the arcsecond level in molecular clouds (Miville-Deschênes et al. 2016).

We add the nebulous contaminant data to our uncontaminated images in the same way we do for fringes, except that there is no 2D polynomial envelope. The whole nebulosity image is background-subtracted (using a SExtractor-like background estimation) to form the final nebulosity pattern $\boldsymbol{N}$ which is then added to the uncontaminated image:

$$\boldsymbol{C} = \boldsymbol{U} + k_{\boldsymbol{N}}\frac{\sigma_{\boldsymbol{U}}}{\sigma_{\boldsymbol{N}}}\boldsymbol{N}, \tag{11}$$

where $k_{\boldsymbol{N}}$ is an empirical scaling factor set to 1.3. The ground truth mask is computed by thresholding $\boldsymbol{N}$ at one sigma above 0. This mask is then eroded with a 6 disk diameter structuring element to remove spurious individual pixels, and dilated with a 22 disk diameter structuring element. An example of added nebulosity is shown in Fig. 2. The light from line-emission nebulae may not necessarily exhibit the same statistical properties as the reflection nebulae targeted for training. However line-emission nebulae are generally brighter and in practice the classifier has no problem detecting them.

2.1.9. Saturation and bleeding (SAT)

Each detector pixel can accumulate only a limited number of electrons. Once the full well limit is reached, the pixel becomes saturated. In CCDs, charges may even *overflow*, leaving saturation trails (a.k.a bleeding trails) along the transfer direction. Such pixels are easily be identified in clean images knowing for each instrument the saturation level.

2.1.10. Diffraction spikes

Diffraction spikes are patterns appearing around bright stars and caused by light diffracting around the spider supporting the secondary mirror. Given the typical cross-shape of spiders, the pattern is usually relatively easy to identify. In some

A&A 634, A48 (2020)

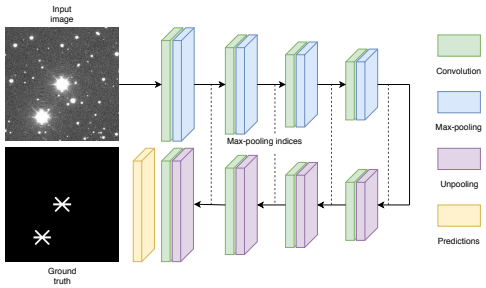


Fig. 3. Neural network used specifically for spike detection.

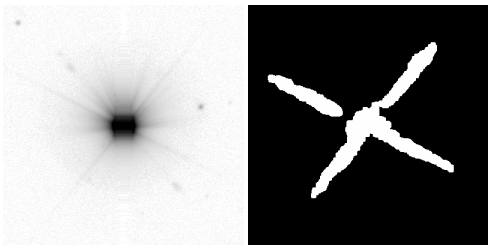


Fig. 4. Example of a spike mask obtained by inference of the separate neural network.

cases, the pattern can deviate significantly from a simple cross because it is affected by various effects, such as distortions, telescope attitude, the truss structure of spider arms, rough edges, or cables around the secondary mirror support, reflections on other telescope structures,... A specific strategy was put in place to build a spikes library to be used to train the CNN.

On the one hand, MegaCam and DECam are mounted on equatorial telescopes and the orientation of spikes is usually (under standard northeast orientation) a "+" for Megacam and an "x" for DECam[1]. On the other hand, HSC is mounted on the alt-az Subaru telescope, and spikes do not display any preferred orientation, making their automated identification more complicated. For this reason, we define a two-step strategy, in which, first, samples of "+"- and "x"-shaped spikes are extracted from DECam and Megacam images, and randomly rotated to generate a library of diffraction spikes with various orientations. The library is then used to train a new CNN that for identifying spikes in HSC images.

MegaCam and DECam analysis. We first identify the brightest stars using SExtractor and extract 300×300 pixel image cutouts around them. The cutouts are thresholded at three sigma above the background and binarized. Element-wise products are computed between these binary images and large "+"-shaped (Megacam) or "x"-shaped (DECam) synthetic masks to isolate the central stars. Each point-wise product is then matched-filtered with a thinner version of the same pattern and binarized using an arbitrary threshold set to 15 ADUs. The empirical size of the spike components is estimated in these masks by measuring the maximum extent of the resulting footprint along any of the two relevant spike directions (horizontal and vertical or diagonals). Finally, the maximum size of the two directions is kept and empirically rescaled to obtain the final spike length and width. If the resulting size is too small, we consider that there is no spike in order to avoid false positives (e.g., a star bright enough to be detected by SExtractor but without obvious spikes). Figure 5 gives an overview of the whole process.

HSC analysis. We train a new neural network to identify spikes in all directions. For that purpose, we build a new training set using the spikes identified in MegaCam and DECam images as described above and apply a random rotation between 0° and 360° to ensure rotational invariance. The neural network has a simple SegNet-like convolutional-deconvolutional architecture (Badrinarayanan et al. 2015), but it is not based on VGG hyper-parameters (Simonyan & Zisserman 2014). It uses 21×21, 11×11, 7×7 and 5×5 convolutional kernels in 8, 16, 32 and 32 feature maps, respectively. The model architecture is shown in Fig. 3. Activation functions are all ELU except on the last layer where it is softmax. It is trained to minimize the softmax cross entropy loss with the Adam optimizer (Kingma & Ba 2014). Each pixel cost is weighted to balance the disproportion between spike and background pixels. If p_s is the spike pixel proportion in the training set, then spike pixels are weighted with $1 - p_s$, while background pixel are weighted with p_s (this is the two-class equivalent of the basic weighting scheme described in Sect. 3.1). Once trained we run inferences on all the brightest stars detected with SExtractor in the HSC images. Output probabilities are binarized based on the MCC (see Eq. (22)) and the resulting mask is empirically eroded and dilated to obtain a clean mask. An example is given in Fig. 4.

2.1.11. Overscan (OV)

Overscan regions are common in CCD exposures, showing up as strips of pixels with very low values at the borders of the frame. To avoid triggering false predictions on real data, overscans must be included in our training set. Doing so, and although these are not truly contaminants, we find it useful to include an "overscan" class in the list of identified features. Overscan regions are simulated by including random strips on the sides of images. Pixel values in the strips are generated in the same way as bad pixel values.

2.1.12. Bright background (BBG) and background (BG)

The objects of interest in this study are the contaminants. Hence, following standard computer vision terminology, all the other types of pixels, including both astronomical objects and empty sky areas, belong to the "background".

We find that defining a distinct class for each of these types of background pixels helps with the training procedure. We thus define the "bright background" (BBG) pixels as pixels belonging to astronomical objects[2] (except nebulosity) present in the uncontaminated images, and background pixels (BG) as pixels covering an empty sky area.

Ground truth masks for bright background pixels are obtained by binarizing the image before adding the contaminants to $10\sigma_U$. The remaining pixels are sky background pixels, which are not affected by any labeled feature.

2.2. Global contaminants

We now describe the data used to identify global contaminants.

[1] DECam images sometimes also exhibit a horizontal spike of unknown origin (Melchior et al. 2016).

[2] Including astrophysical sources in the "background" class can seem somewhat counter-intuitive in a purely astronomical context, but for consistency we choose to follow the computer vision terminology and meaning.

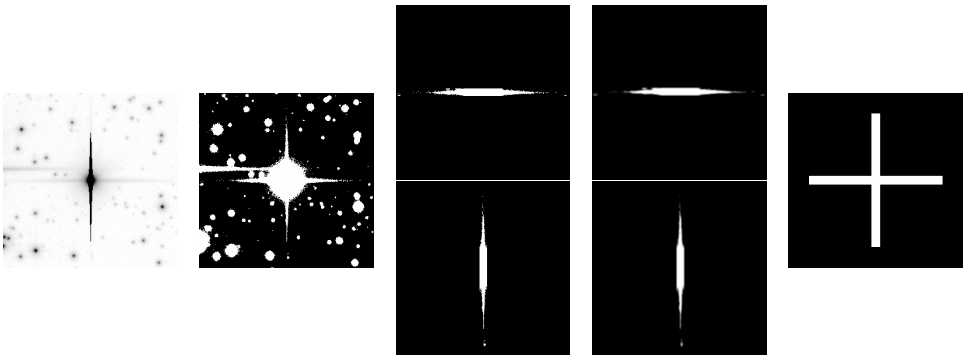


Fig. 5. Empirical spike flagging process. *From left to right*: source image centered on a bright star candidate, the same image thresholded, the two pointwise products, the matched filtered pointwise products, the final mask drawed from the empirical size computed with the two previous masks.

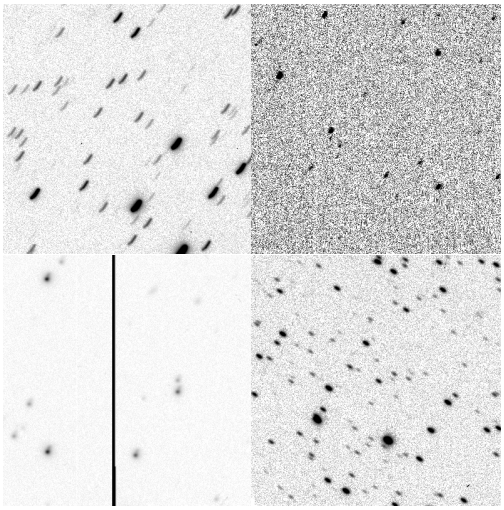


Fig. 6. Examples of images affected by tracking errors.

Tracking errors happen when the telescope moves during an exposure due to, for instance, telescope guiding or tracking failures, wind gusts, or earthquakes. As illustrated in Fig. 6, this causes all the sources to be blurred along a path on the celestial sphere generated by the motion of the telescope. Because tracking errors affect the entire focal plane, the analysis is performed globally on the whole image. The library of real images affected by TR events is a compilation of exposures identified in the COSMIC-DANCE survey for the cameras of Table 1, and images that were gathered over the years at the UKIRT telescope, kindly provided to us by Mike Read.

2.3. Generating training samples

Both types of contaminants – global and local contaminants – must be handled separately: they require different neural network architectures, and different training data sets as well.

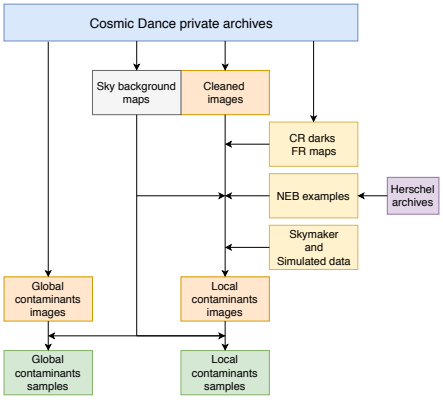


Fig. 7. Schematic view of the sample production pipeline. All COSMIC-DANCe archive images have their background map computed. Clean images are built from the COSMIC-DANCe archives. Contaminants from diverse sources (COSMIC-DANCe archives, *Herschel* archives or simulations) are added to clean images; this step uses the background maps. The resulting local contaminant images are dynamically compressed (see Sect. 2.3.3) and ready to be fetched into the neural network. Global contaminant samples are directly obtained from the COSMIC DANCe archives and dynamically compressed.

Figure 7 gives a synthetic view of the sample production pipeline and the various data sources.

The breakdown per imaging instrument of the COSMIC DANCe dataset is listed Table 3.

The following subsections treat about some special features of the sample generation.

2.3.1. Local contaminants

The order in which local contaminants are added is important. Bad columns, lines, and pixels are added last because they are static defaults defining the final value of a pixel, no matter how many photons hit them.

In our neural network architecture contaminant classes do not need to be mutually exclusive. Each pixel can be assigned several classes as several defaults can affect a given pixel

A&A 634, A48 (2020)

Table 3. COSMIC-DANCE archive usage per imaging instrument.

Instrument	Clean	CR	No TR	TR
DECam	✓		✓	
MOSAIC2		✓		
MOSAIC1		✓		
NEWFIRM			✓	✓
Megacam	✓	✓	✓	✓
CFH12K		✓	✓	✓
CFH8K				✓
WFC			✓	✓
WFCAM			✓	✓
Direct CCD (LCO Swope)			✓	✓
VST		✓	✓	✓
HSC	✓	✓	✓	
VIRCAM			✓	✓

Notes. Clean is for uncontaminated images, CR for dark images used for cosmic ray identification, No TR is for images not affected by tracking errors, and TR for images affected by tracking errors.

Table 4. All the contaminants and their abbreviated names.

Contaminant	Abbreviation
Cosmic rays	CR
Hot columns/lines	HCL
Dead columns/lines/clusters	DCL
Hot pixels	HP
Dead pixels	DP
Persistence	P
Trails	TRL
Fringes	FR
Nebulosities	NEB
Saturated pixels	SAT
Diffraction spikes	SP
Overscanned pixels	OV
Bright background	BBG
Background	BG

(e.g., fringes and cosmic ray hit). On the other hand, the faint background class that defines pixels not affected by any default excludes all other classes. A list of all the contaminants included in this study are presented in Table 4.

Figure 8 shows examples of local contaminant sample input images, each with its color-coded ground truth.

2.3.2. Global contaminants

The global contaminant dataset contains images that have been hand labeled as affected by tracking errors or not. The images, taken from the COSMIC DANCe archives, are not cleaned, hence they are potentially affected by preexisting local contaminants. This is because the global contaminant detector is intended to be operated before the local one.

2.3.3. Dynamic compression

All images are dynamically compressed before being fed to the neural networks using the following procedure:

$$\tilde{\boldsymbol{C}} = \operatorname{arsinh}\left(\frac{\boldsymbol{C} - \boldsymbol{B} + \mathcal{N}(0, \sigma_U^2)}{\sigma_U}\right).$$

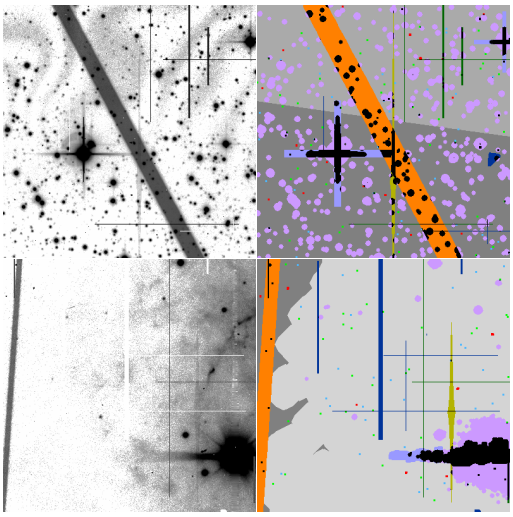


Fig. 8. Examples of input (*left*) and their ground truth (*right*). Each class is assigned a color so that the ground truth can be represented as a single image (red: CR, dark green: HCL, dark blue: BCL, green: HP, blue: BP, yellow: P, orange: TRL, gray: FR, light gray: NEB, purple: SAT, light purple: SP, brown: OV, pink: BBG, dark gray: BG). Pixels that belong to several classes are represented in black. In the interest of visualization, hot and dead pixel masks have been morphologically dilated so that they appear as 3×3 pixel areas in this representation.

The aim of dynamic compression is to reduce the dynamic range of pixel values, which is found to help neural network convergence. The image is first background subtracted. Then, a small random offset is added to increase robustness regarding background subtraction residuals. The resulting image is normalized by the standard deviation of the background noise and finally compressed through the arsinh function, which has the property to behave linearly around zero and logarithmically for large (positive or negative) values.

2.3.4. Data augmentation

We deploy data augmentation techniques to use our data to the maximum of its information potential. The two following data augmentation procedures are applied to the set of local contaminant training samples. First, random rotations, using as angles multiples of 90°, are applied to cosmic ray, fringe patterns, and nebulosity patterns. Secondly, some images are rebinned. When picking up a clean image, we check if the image can be 2×2 rebinned with the constraint that the FWHM remains greater than 2 pixels – the FWHM of the image was previously estimated using SExtractor (Bertin & Arnouts 1996). This value is chosen on the basis of the plate sampling offered by current ground-based imagers. If the image can be 2×2 rebinned while meeting the condition above, it has a 50% probability to be rebinned.

3. Convolutional neural networks

In this section, we describe the convolutional neural networks used for our analysis. The first one, MaxiMask, classifies pixels ("local contaminants") while the second one, MaxiTrack, classifies images ("global contaminants").

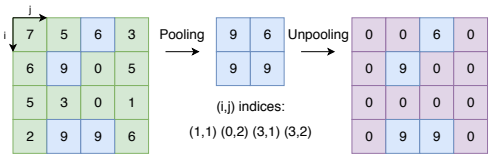


Fig. 9. Example of an unpooling process. Indices of max-pooling are kept up and reused to upsample the feature maps.

3.1. Local contaminant neural network

3.1.1. Architecture

The model used for the semantic segmentation of the local contaminants, MaxiMask, is based on Badrinarayanan et al. (2015) and Yang et al. (2018), which both rely on a VGG-like architecture (Simonyan & Zisserman 2014). It consists of three parts.

The first part contains single and double convolutional layers followed by max-pooling downsampling. This enables the network to compute relevant feature maps at different scales. During this step, max-pooling pixel indices are kept up for later reuse.

The second part also incorporates convolutional layers and recovers spatial resolution by upsampling feature maps using the max-pooling indices. An example of unpooling is given in Fig. 9. At each resolution level, the feature maps of the first part are summed with the corresponding upsampled feature maps to make use of the maximum of information.

The third part is made of extra unpool-convolution paths (UCPs) that recover the highest image resolution from each feature map resolution so that the network can exploit the maximum of information of each resolution. Thus, it results 5 pre-predictions, one for each resolution.

The 5 pre-predictions are finally concatenated and a last convolution layer builds the final predictions. The sigmoid activation functions in this last layer are not *softmax*-normalized, to allow non-mutually exclusive classes to be assigned jointly to pixels. All convolutional layers use 3×3 kernels and apply ReLU activations. The architecture is represented in Fig. 10 and hyperparameters are described more precisely in Table 5. The neural network is implemented using the TensorFlow library (Abadi et al. 2016) on a TITAN X Nvidia GPU.

3.1.2. Training and loss function

Training is done for 30 epochs on 50 000 images, with mini-batches shuffled at every epoch. The batch size is kept small (10) to maintain a reasonable memory footprint. The model is trained end-to-end using the Adam optimizer (Kingma & Ba 2014). The loss function L is the sigmoid cross-entropy (Rubinstein 1999) summed over all classes and pixels, and averaged across batch images:

$$L = -\frac{1}{\mathrm{card}(\mathcal{B})} \sum_{b\in\mathcal{B}} \sum_{p\in\mathcal{P}} w'_{p,b} \sum_{\omega_c\in C} \Big(y_{b,p,c} \log \hat{y}_{b,p,c} + (1 - y_{b,p,c}) \log(1 - \hat{y}_{b,p,c}) \Big), \tag{12}$$

where $\mathcal{B}$ is the set of batch images, $\mathcal{P}$ is the set of all image pixels, C is the set of all contaminant classes, $w'_{p,b}$ is a weight applied to pixel p of image b in the batch (see below), $\hat{y}_{b,p,c}$ is the sigmoid prediction for class ω_c of pixel p of image b in the batch, and $y_{b,p,c}$ is the ground truth label for class ω_c of pixel p of image b defined as:

$$y_{b,p,c} = \begin{cases} 1 & \text{if } \omega_c \in C_{p,b} \\ 0 & \text{otherwise} \end{cases}, \tag{13}$$

where $C_{p,b} \subset C$ is the set of contaminant classes labeling pixel p of image b in the batch. In order to improve the back-propagation of error gradients down to the deepest layers, several losses are combined. In addition to the main sigmoid cross-entropy loss L computed on the final predictions, we can compute a sigmoid cross-entropy for each of the 5 pre-predictions. There are several ways to associate all of these losses. Like Yang et al. (2018), we find that adding respectively 33% or 50% of each of the 3 or 2 smallest resolution losses to the main loss works best. The two main rules here are that the additional loss weights should sum to 1 and that higher resolution pre-predictions become less informative as they get closer to the one at full resolution.

Basic training procedures are vulnerable to strong class imbalance, which makes it more likely for the neural network to converge to a state where rare contaminants are not properly detected. Contaminant classes are so statistically insignificant (down to one part in 10^6 with real data, typically) that the classifier may be tricked into assigning all pixels to the background class. To prevent this, we start by applying a basic weighting scheme to each pixel according to its class representation in the training set, that is each pixel p of batch image b belonging to classes in $C_{p,b}$ is weighted by $w_{p,b}$ defined as

$$w_{p,b} = \sum_{\omega_c\in C_{p,b}} w_c, \tag{14}$$

with

$$w_c = \left(P(\omega_c|T) \sum_i \frac{1}{P(\omega_i|T)} \right)^{-1}, \tag{15}$$

where $P(\omega_c|T)$ is the fraction of pixels labeled with class ω_c in the training dataset T. The $P(\omega_c|T)$'s do not sum to one as many pixels belong to several classes and are thus counted several times. We find that the weighting scheme brings slightly better results and less variability in the training if weights are computed at once from the class proportions of the whole set, instead of being recomputed for each image. From Eq. (15) we have:

$$\forall i \in C, \forall j \in C, \ \frac{w_i}{w_j} = \frac{P(\omega_j|T)}{P(\omega_i|T)} \ \text{ and } \sum_{\omega_c\in C} w_c = 1. \tag{16}$$

However, with this simple weighting scheme, background class pixels that are close to rare features are given very low weights, although they are decisive for classification. To circumvent this, weight maps are smoothed with a 3×3 Gaussian kernel with unit standard deviation so that highly weighted regions spread over larger areas. Other kernel sizes and standard deviations were tested but we find 3 and 1 to give the best results. The resulting weights of this smoothing are the $w'_{p,b}$ presented in the loss function of Eq. (12).

Finally, the solution is regularized by the l2 norm of all the N network weights, by adding the following term to the total loss:

$$L2_{\mathrm{reg}} = \lambda \sum_i^N \|\boldsymbol{k}_i\|_2, \tag{17}$$

where the $\boldsymbol{k}_i$'s are the convolution kernel vectors. λ sets the regularization strength. We find $\lambda = 1$ to provide the best results.

A&A 634, A48 (2020)

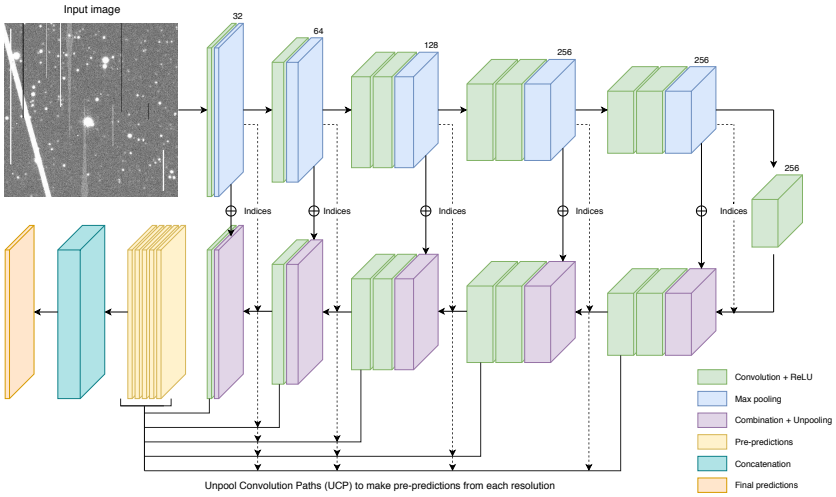


Fig. 10. Scheme representation of the local contaminants neural network architecture.

Table 5. Description of the local contaminants neural network architecture, including map dimensions.

Layer	Size	UCP from each resolution			
Input	$400\times400\times1$				
Conv	$400\times400\times32$				
Maxpool	$200\times200\times32$				
Conv	$200\times200\times64$				
Maxpool	$100\times100\times64$				
Conv	$100\times100\times128$				
Conv	$100\times100\times128$				
Maxpool	$50\times50\times128$				
Conv	$50\times50\times256$				
Conv	$50\times50\times256$				
Maxpool	$25\times25\times256$				
Conv	$25\times25\times256$				
Conv	$25\times25\times256$				
Maxpool	$13\times13\times256$				
Conv	$13\times13\times256$				
Unpooling	$25\times25\times256$				
Conv	$25\times25\times256$				
Conv	$25\times25\times256$	UCP			
Unpooling	$50\times50\times256$	Idem			
Conv	$50\times50\times256$	None			
Conv	$50\times50\times128$	Idem	UCP		
Unpooling	$100\times100\times128$	Idem	Idem		
Conv	$100\times100\times128$	None	None		
Conv	$100\times100\times64$	Idem	Idem	UCP	
Unpooling	$200\times200\times64$	Idem	Idem	Idem	
Conv	$200\times200\times32$	Idem	Idem	Idem	UCP
Unpooling	$400\times400\times32$	Idem	Idem	Idem	Idem
Conv	$400\times400\times14$	Idem	Idem	Idem	Idem
Concat		$400\times400\times70$			
Conv		$400\times400\times14$			

Notes. All convolution kernels are 3×3 and max-pooling kernels are 2×2. All activation functions (not shown for brevity) are ReLU, except in the output layer where the sigmoid is used.

3.2. Global contaminant neural network architecture

The convolutional neural network that detects global contaminants (tracking errors), MaxiTrack, is a simple network made of convolutional layers followed by max-pooling and fully connected layers. The architecture of the network is schematized in Fig. 11 and detailed in Table 6. Because the two classes are mutually exclusive (affected by tracking errors or not), we adopt for the output layer a softmax activation function and a softmax cross-entropy loss function (Rubinstein 1999). Training is done for 48 epochs on 50 000 images with a mini-batch size of 64 samples, using the Adam optimizer.

4. Results with test data and quality assessment

4.1. Local contaminants neural network

We evaluate the quality of the results in several ways. First, we estimate the performance of the network on test data, both quantitatively through various metrics, and qualitatively. We verify that there is no over-fitting by checking that performance on the test set is comparable to that on the training set. Next, we show that performance is immune to the presence or absence of other contaminants in a given image. We finally compare the performance of the cosmic ray detector to that of a classical algorithm.

4.1.1. Performance metrics

We first estimate classification performance on a benchmark test set comprising 5000 images. Because the network is a binary classifier for every class, we can compute a Receiver Operating Characteristic (ROC) curve for each of them. ROC curves represent the True Positive Rate (TPR) vs. the False Positive Rate (FPR):

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN}, \tag{18}$$

$$FPR = \frac{FP}{N} = \frac{FP}{TN + FP}, \tag{19}$$

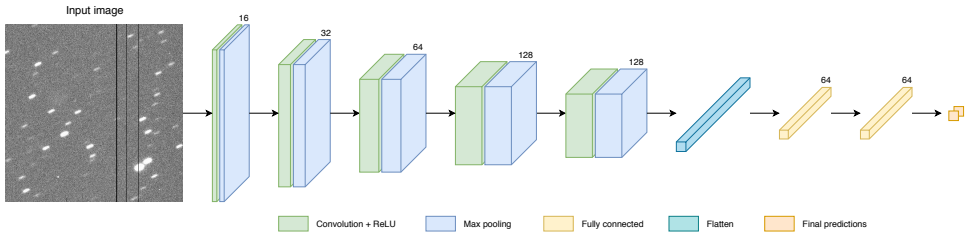


Fig. 11. Scheme representation of the global contaminants neural network architecture.

Table 6. Description of the global contaminant neural network architecture, including map dimensions.

Layer	Size
Input	$400 \times 400 \times 1$
Conv	$400 \times 400 \times 16$
Maxpool	$200 \times 200 \times 16$
Conv	$200 \times 200 \times 32$
Maxpool	$100 \times 100 \times 32$
Conv	$100 \times 100 \times 64$
Maxpool	$50 \times 50 \times 64$
Conv	$50 \times 50 \times 128$
Maxpool	$25 \times 25 \times 128$
Conv	$25 \times 25 \times 128$
Maxpool	$13 \times 13 \times 128$
Flatten	21 632
Fully connected	64
Fully connected	64
Fully connected	2

Notes. All convolution kernels are 9×9 and max-pooling kernels are 2×2. All activation functions (not shown for brevity) are ReLU, except in the output layer where predictions are done using softmax.

where P is the number of contaminated pixels, TP is the number of true positives (contaminated pixels successfully recovered as contaminated), FN is the number of false negatives (contaminated pixels wrongly classified as non-contaminated), N is the number of non-contaminated pixels, FP is the number of false positives (non-contaminated pixels wrongly classified as contaminated), and TN is the number of true negatives (non-contaminated pixels successfully recovered as non-contaminated).

The accuracy (ACC) is subsequently defined as

$$ACC = \frac{TP + TN}{P + N}. \tag{20}$$

The more the ROC curve bends toward the upper left part of the graph, the better the classifier. However with strongly imbalanced datasets, such as our pixel data, one must be very cautious with the TPR, FPR and ACC values for assessing the quality of the results. For example, if one assumes that there are 1000 pixels of the contaminant class (P) and 159 000 pixels of the background class (N) in a 400×400 pixel sub-image, a TPR of 99% and a FPR of 1%, corresponding to an accuracy of 99%, would actually represent a poor performance, as it would imply 990 true positives, 10 false negatives, 157 410 true negatives, and 1590 false positives. In the end, there would be more false positives FP (pixels wrongly classified as contaminated) than true positives TP.

For this reason the ROC curves in Fig. A.1 are displayed with a logarithmic scale on the FPR axis. We require the FPR to be very low (e.g smaller than 10^{-3}) to consider that the network performs properly.

On the other hand, recovering the exact footprint of large, fuzzy defects is almost impossible at the level of individual pixels, which makes the classification performance for persistence, satellite trails, fringes, nebulosities, spikes and background classes look worse in Fig. A.1 than it really is in practice.

Also, two ROC curves are drawn for cosmic rays and trails. The second one (in green) is computed using only the instances of the class that are above a specific level of the sky background. These instances were defined by retaining those which had more than a half of their pixels above 3σ. These second curves shows that the network performs better on more obvious cases.

In addition to the FPR, TPR, ACC and AUC, we use two other metrics helpful for assessing the network performance: the purity (or precision), representing the fraction of correct predictions among the positively classified samples, and the Matthews correlation coefficient (MCC, Matthews 1975), which is an accuracy measure that takes into account the strong imbalance between classes.

$$PUR = \frac{TP}{TP + FP} = \text{Purity or Precision}, \tag{21}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \tag{22}$$

In the above example, the purity would reach only 38% and the MCC only 61%, highlighting the classifier poor positive class discrimination.

Figure A.3 shows the true positive rate against the purity. Again, the purple curve represents how a random classifier would perform. In these curves the best classifier would sit in the top right ($TPR = 1$ and $PUR = 1$). The darkest points also represent lowest thresholds while the lighter are the highest ones.

Some qualitative results are presented in Fig. 12. A given pixel is assigned a given class if its probability to belong to this class is higher than the best threshold in the sense of the MCC.

Finally, MCCs are represented in Fig. A.2, as a function of the output threshold. In each curve, the threshold giving the best MCC is annotated around the best MCC point. It is important to note that the best threshold depends on the modification of the prior that has been applied to the raw output probabilities. This update of the prior is explained in Sect. 5.

4.1.2. Robustness regarding the context

The MaxiMask neural network is trained using mostly images that include all contaminant classes. Hence, we must check if the network performs equally well independently of the context,

A&A 634, A48 (2020)

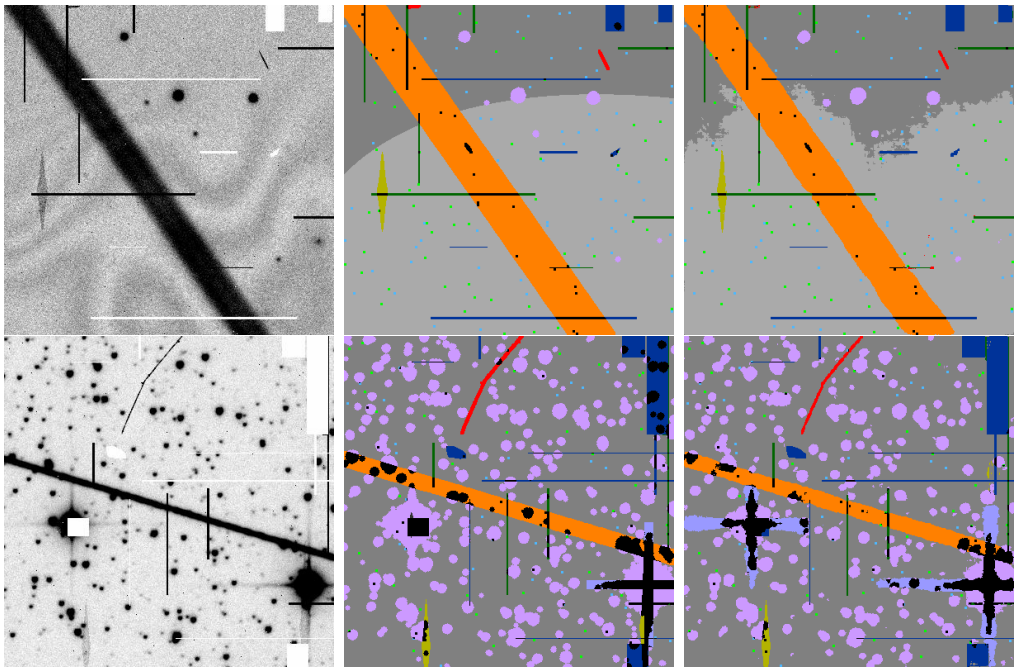


Fig. 12. Examples of qualitative results on test data. *Left*: input; *middle*: ground truth; *right*: predictions. Each class is assigned a color so that the ground truth can be represented in one single image. Class predictions are done according to the threshold giving the highest MC coefficient. The color coding is identical to that of Fig. 8.

that is if it delivers equally good results for images containing, for example, a single class of contaminant.

To this aim, for every contaminant class, we generate a dataset of 1000 images affected only by this type of contaminant (except saturated and background pixels), and another dataset of 1000 images containing only saturated and background pixels. We then compare the performance of MaxiMask for each class with the that obtain on the corresponding dataset. We find that performance (AUC) is similar or even slightly higher for the majority of the classes. This shows that the network is not conditioned to work only in the exact context of the training. The results are presented in Table 7.

As it can be seen, for all classes but fringes and nebulosity, performance improves when a single type of contaminant is present. The slight improvement may come from the fact that ambiguous cases (when pixels are affected by more than one contaminant class, e.g., a cosmic ray or a hot pixel over a satellite trail) are not present in the single contaminant test set.

Table 7. AUC of each class depending on the test set context.

Class	All contaminant set AUC	Single contaminant set AUC
CR	0.96927	0.98314
HCL	0.99763	0.99957
DCL	0.99872	0.99976
HP	0.99741	0.99965
DP	0.99739	0.99975
P	0.99352	0.99951
TRL	0.99511	0.99813
FR	0.98057	0.93326
NEB	0.97895	0.84575
SAT	0.99965	0.99974
SP	0.96125	0.98061
OV	0.99997	1.00000
BBG	0.98484	0.99165
BG	0.96895	0.98371

4.1.3. Cosmic rays: effect of PSF undersampling and comparison with LA Cosmic

Undersampling makes cosmic ray hits harder to distinguish from point-sources. To solve this issue, van Dokkum (2001) has developed LA Cosmic, a method based on a variation of Laplacian edge detection. It is largely insensitive to cosmic ray morphology and PSF sampling. LA Cosmic thus offers an excellent opportunity to test the performance of MaxiMask on undersampled exposures.

To do so, we generate two datasets containing only the cosmic ray contaminant class (plus object and background). A well sampled set of images with FWHMs larger than 2.5 pixels, and an undersampled image set with FWHMs smaller than 2.5 pixels. We run MaxiMask and the Astro-SCRAPPY Python implementation LA Cosmic. To make a fair comparison, LA Cosmic masks are dilated in the same way as the ground truth cosmic ray masks of MaxiMask. However, while MaxiMask

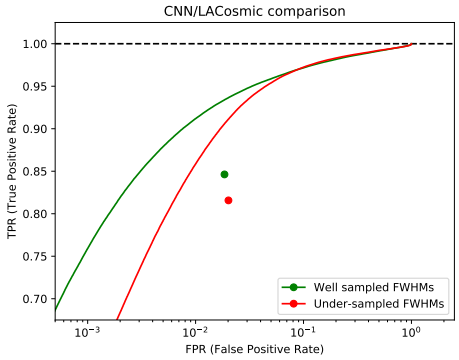


Fig. 13. CR detection performance comparison with LA Cosmic.

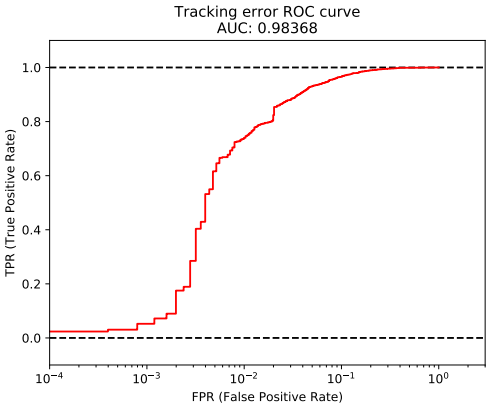


Fig. 14. Global contaminant neural network ROC curve; the steps are a consequence of limited statistics.

generates probability maps that can be thresholded at different levels, LA Cosmic only outputs a binary mask. To compare the results we therefore build ROC curves for the neural network and over-plot a single point representing the result obtained with LA Cosmic.

Figure 13 shows that the neural network performs better than LA Cosmic in both regimes with our data.

4.2. Global contaminants neural network

The ROC curve for the global contaminant neural network is shown in Fig. 14. It is computed from a test set of 5000 images.

5. Modifying priors

If one knows what class proportions are expected in the observation data, output probabilities can be updated to better match these priors (e.g., Saerens et al. 2002; Bailer-Jones et al. 2008).

The outputs of a *perfectly trained* neural network classifier with a cross-entropy loss function can be interpreted as Bayesian posterior probabilities (e.g., Richard & Lippmann 1991; Hampshire & Pearlmutter 1991; Rojas 1996). Under this assumption and using Bayes' rule, the output for the class ω_c of the trained neural network model defined by a training set T writes:

$$P(\omega_c|\,\boldsymbol{x}, T) = \frac{p(\boldsymbol{x}|\omega_c, T)P(\omega_c|T)}{\sum\limits_{\omega\in\{\omega_c,\bar{\omega_c}\}} p(\boldsymbol{x}|\omega, T)P(\omega|T)}, \tag{23}$$

where $\boldsymbol{x}$ is the input image data around the pixel of interest, $p(\boldsymbol{x}|\omega_c, T)$ is the distribution of $\boldsymbol{x}$ conditional to class ω_c in the training set T, and $P(\omega_c|T)$ is the prior probability of a pixel to belong to the class ω_c in the trained model.

As each output acts as a binary classifier, the sum is done on the class ω_c (contaminant) and its complementary $\bar{\omega_c}$ ("not the contaminant").

With the observation data set O we may similarly write:

$$P(\omega_c|\boldsymbol{x}, O) = \frac{p(\boldsymbol{x}|\omega_c, O)P(\omega_c|O)}{\sum\limits_{\omega\in\{\omega_c,\bar{\omega_c}\}} p(\boldsymbol{x}|\omega, O)P(\omega|O)}, \tag{24}$$

where $P(\omega_c|O)$ is the expected fraction of pixels with class ω_c in O.

Now, if the appearance of defects in O matches that in the training set T, we have $p(\boldsymbol{x}|\omega_c, T) = p(\boldsymbol{x}|\omega_c, O)$, and we can rewrite (24) as:

$$P(\omega_c|\boldsymbol{x}, O) = \frac{P(\omega_c|\boldsymbol{x}, T)\frac{P(\omega_c|O)}{P(\omega_c|T)}}{\sum\limits_{\omega\in\{\omega_c,\bar{\omega_c}\}} P(\omega|\boldsymbol{x}, T)\frac{P(\omega|O)}{P(\omega|T)}} \tag{25}$$

$$= \frac{1}{1+\left(\frac{1}{P(\omega_c|\boldsymbol{x},T)}-1\right)\frac{P(\omega_c|T)}{P(\omega_c|O)}\frac{1-P(\omega_c|O)}{1-P(\omega_c|T)}}. \tag{26}$$

If pixels were all weighted equally, the training priors $P(\omega_c|T)$ would simply be the class proportions in the training set. However, this is not the case here, and pixel weights have to be taken into account. To do so, we follow Bailer-Jones et al. (2008)'s approach, by using as an estimator of $P(\omega_c|T)$ the posterior mean on the test set T' (which by construction is distributed identically to the training set):

$$\hat{P}(\omega_c|T) = \frac{1}{\mathrm{card}(T')}\sum_{\boldsymbol{x}\in T'} P(\omega_c|\boldsymbol{x}, T'). \tag{27}$$

These corrected probabilities are used to compute the MC coefficient curves in Fig. A.2 (whereas the prior correction does not affect the ROC and purity curves).

MaxiMask comes with the $P(\omega_c|T)$ values already set, therefore one only needs to specify the expected class proportions in the data, that is the $P(\omega_c|O)$'s.

6. Application to other data

As a sanity check, we apply MaxiMask to data obtained from different instruments not part of the training set. Examples of the resulting contaminant maps are shown in appendix.

Our first external check is with ZTF (Bellm et al. 2019) data. The MaxiMask output for a science image featuring a prominent trail with variable amplitude is shown in Fig. A.4. We can note the ability of MaxiMask to properly flag both the trail and overlapping sources.

Our second external check is with the ACS instrument onboard the *Hubble* Space Telescope (Fig. A.5 and A.6). This test illustrates MaxiMask's ability to distinguish cosmic rays from poorly sampled, diffraction-limited point source images.

A&A 634, A48 (2020)

Given the seemingly good performance of MaxiMask on images from instruments not part of the training set, one question that may arise is whether MaxiMask can readily be used on production for such instruments, without any retraining or transfer learning. Our limited experience with MaxiMask seems to indicate that this is indeed the case, although retraining may be beneficial for specific instrumental features. As shown here, excellent performance can be reached by training with 50 000 400×400 images taken from three different instruments. We think that a minimum of 10 000 400×400 would be a good start to train on a single instrument. Assuming CCDs of approximately 2000×2000 pixels, thus containing $25\,400 \times 400$ images, it would just need 400 CCDs, equivalent to 10 fields for a 40 CCD camera.

Our last series of tests is conducted on digital images of natural scenes (landscape, cat, human face), to check for possible inconsistencies on data that are totally unlike those from the training set. Reassuringly, the maps produced by MaxiMask are consistent with the expected patterns. For instance, the cat's whiskers are identified as cosmic ray impacts, and pixels with the lowest values as bad pixels.

7. Using MaxiMask and MaxiTrack

MaxiMask and MaxiTrack are available online[3]. MaxiMask is a Python module that infers probability maps from FITS images. It can process a whole mosaic, a specific FITS image extension, or all the FITS files from a directory or a file list. For every FITS file being processed a new FITS image is generated with the same HDU (Header Data Unit) structure as the input. Every input image HDU has a matching contaminant map HDU in output, with one image plane per requested contaminant. The header contains metadata related to the contaminant, including the prior and threshold used. An option can be set to generate a single image plane for all contaminants, using a binary code for each contaminant. Such composite contaminant maps can easily be used as flag maps, for example, in SExtractor. Based on command line arguments and configuration parameters, one can select specific classes, apply updates to the priors and thresholds to the probability maps. The code relies on the TensorFlow library and can work on both CPUs or GPUs, although the CPU version is expected to be much slower: MaxiMask processes about 1.2 megapixel per second with an NVidia Titan X GPU, and about 60 times less on a 2.7 GHz Intel i7 dual-core CPU. Yet, there is probably room for improvement in processing efficiency for both the CPU and GPU versions.

MaxiTrack is used the same way as MaxiMask, except that the output is a text file indicating the probability for the input image(s) to be affected by tracking errors (one probability per extension if the image contains several HDUs). It can also apply an update to the prior. It runs at 60 megapixels s^{-1} with an NVidia Titan X GPU and is 9 times slower on a 2.7 GHz Intel i7 dual-core CPU.

8. Summary and perspectives

We have built a data set and trained convolutional neural network classifiers named MaxiMask and MaxiTrack to identify contaminants in astronomical images. We have shown that they achieve good performance on test data, both real and simulated. By delivering posterior probabilities, MaxiMask and MaxiTrack give the user the flexibility to set appropriate threshold levels and achieve the desired TPR/FPR trade-offs depending on the scientific objectives and requirements. Both classifiers require no input parameters or knowledge of the camera properties.

Even though the mix of contaminants in the training set is unrealistic, being dictated by training requirements, we have checked that this does not impact performance. Output probabilities can be corrected to adapt the behavior of MaxiMask to any mix of contaminants in the data.

We are aware that several types of contaminants and images are missing from the current version and may be added in the future.

Local contaminants include two particularly prominent classes of contaminants: optical and electronics ghosts. Unwanted reflections within the optics result in stray light in exposures. These reflections can produce spurious images from bright sources commonly referred to as "optical ghosts". Sometimes, reflections from very bright stars outside of the field may also be seen. Detectors read through multiple ports also suffer from a form of electronic ghost known as cross-talk. Electronic cross-talk causes bright sources in one of the CCD quadrants to generate a ghost pattern in other quadrants. The ghosts may be negative or positive and are typically at the level of $1{:}10^4$. Both effects are a significant source of nuisance in wide field exposures, especially in crowded fields and deep images, where they generate false, transient sources, and can affect high precision astrometric and photometric measurements.

Another category of common issues is defocused or excessively aberrated exposures, as well as trails caused by charge transfer inefficiency, all of which which could easily be implemented in MaxiTrack.

Also, the training set used in the current version of MaxiMask and MaxiTrack does not include images from space-born telescopes nor, more generally, diffraction-limited imagers. Therefore, they are unlikely to perform optimally with such data, although limited testing indicates that they may remain usable for most features, an example of prediction on HST data is shown in Figs. A.5 and A.6.

Finally, MaxiMask could be extended to not only detect contaminants, but also to generate an inpainted (i.e., "corrected") version of the damaged image areas wherever possible.

Acknowledgements. M. P. acknowledges financial support from the Centre National d'Etudes Spatiales (CNES) fellowship program. We are grateful to Mike Read, of the Royal Observatory, Edinburgh, for providing us with data from the UKIRT telescope, and to Vincent Lepetit for providing comments and suggestions that helped improve the paper. This research has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 682903, P.I. H. Bouy), and from the French State in the framework of the "Investments for the future" Program, IdEx Bordeaux, reference ANR-10-IDEX-03-02. We gratefully acknowledge the support of NVIDIA Corporation with the donation of one of the Titan Xp GPUs used for this research. This research draws upon data distributed by the NOAO Science Archive. NOAO is operated by the Association of Universities for Research in Astronomy (AURA) under cooperative agreement with the National Science Foundation. Based on observations made with the *Isaac Newton* Telescope operated on the island of La Palma by the Isaac Newton Group in the Spanish Observatorio del Roque de los Muchachos of the Instituto de Astrofísica de Canarias. The *Isaac Newton* Telescope is operated on the island of La Palma by the Isaac Newton Group in the Spanish Observatorio del Roque de los Muchachos of the Instituto de Astrofísica de Canarias. This paper makes use of data obtained from the Isaac Newton Group Archive which is maintained as part of the CASU Astronomical Data Centre at the Institute of Astronomy, Cambridge. Based on data obtained from the ESO Science Archive Facility. This research used the facilities of the Canadian Astronomy Data Centre operated by the National Research Council of Canada with the support of the Canadian Space Agency. Based in part on data collected at Subaru Telescope which is operated by the National Astronomical Observatory of Japan and obtained from the SMOKA,

[3] https://www.github.com/mpaillassa/MaxiMask

which is operated by the Astronomy Data Center, National Astronomical Observatory of Japan. The Hyper Suprime-Cam (HSC) collaboration includes the astronomical communities of Japan and Taiwan, and Princeton University. The HSC instrumentation and software were developed by the National Astronomical Observatory of Japan (NAOJ), the Kavli Institute for the Physics and Mathematics of the Universe (Kavli IPMU), the University of Tokyo, the High Energy Accelerator Research Organization (KEK), the Academia Sinica Institute for Astronomy and Astrophysics in Taiwan (ASIAA), and Princeton University. Funding was contributed by the FIRST program from Japanese Cabinet Office, the Ministry of Education, Culture, Sports, Science and Technology (MEXT), the Japan Society for the Promotion of Science (JSPS), Japan Science and Technology Agency (JST), the Toray Science Foundation, NAOJ, Kavli IPMU, KEK, ASIAA, and Princeton University. This paper makes use of software developed for the Large Synoptic Survey Telescope. We thank the LSST Project for making their code available as free software at http://dm.lsst.org. The Pan-STARRS1 Surveys (PS1) have been made possible through contributions of the Institute for Astronomy, the University of Hawaii, the Pan-STARRS Project Office, the Max-Planck Society and its participating institutes, the Max Planck Institute for Astronomy, Heidelberg and the Max Planck Institute for Extraterrestrial Physics, Garching, The Johns Hopkins University, Durham University, the University of Edinburgh, Queen's University Belfast, the Harvard-Smithsonian Center for Astrophysics, the Las Cumbres Observatory Global Telescope Network Incorporated, the National Central University of Taiwan, the Space Telescope Science Institute, the National Aeronautics and Space Administration under Grant No. NNX08AR22G issued through the Planetary Science Division of the NASA Science Mission Directorate, the National Science Foundation under Grant No. AST-1238877, the University of Maryland, and Eotvos Lorand University (ELTE) and the Los Alamos National Laboratory. Based on data collected at the Subaru Telescope and retrieved from the HSC data archive system, which is operated by Subaru Telescope and Astronomy Data Center at National Astronomical Observatory of Japan. This paper includes data gathered with the Swope telescope located at Las Campanas Observatory, Chile. Based on observations obtained with MegaPrime/MegaCam, a joint project of CFHT and CEA/DAPNIA, at the Canada-France-Hawaii Telescope (CFHT) which is operated by the National Research Council (NRC) of Canada, the Institut National des Science de l'Univers of the Centre National de la Recherche Scientifique (CNRS) of France, and the University of Hawaii. This research has made use of NASA's Astrophysics Data System Bibliographic Services. This research made use of Astropy, a community-developed core Python package for Astronomy (Astropy Collaboration, 2013, http://dx.doi.org/10.1051/0004-6361/201322068). The *Herschel* spacecraft was designed, built, tested, and launched under a contract to ESA managed by the *Herschel*/Planck Project team by an industrial consortium under the overall responsibility of the prime contractor Thales Alenia Space (Cannes), and including Astrium (Friedrichshafen) responsible for the payload module and for system testing at spacecraft level, Thales Alenia Space (Turin) responsible for the service module, and Astrium (Toulouse) responsible for the telescope, with in excess of a hundred subcontractors

References

Abadi, M., Barham, P., Chen, J., et al. 2016, in Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16), 16, 265
Autry, R. G., Probst, R. G., Starr, B. M., et al. 2003, in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, eds. M. Iye, & A. F. M. Moorwood, 4841, 525
Badrinarayanan, V., Kendall, A., & Cipolla, R. 2015, ArXiv e-prints [arXiv:1511.00561]
Badrinarayanan, V., Kendall, A., & Cipolla, R. 2017, IEEE Trans. Pattern Anal. Mach. Intell., 39, 2481
Bailer-Jones, C. A., Smith, K., Tiede, C., Sordo, R., & Vallenari, A. 2008, MNRAS, 391, 1838
Bekteševi ć, D., Vinković, D., Rasmussen, A., & Ivezić, Ž. 2018, MNRAS, 474, 4837
Bellm, E. C., Kulkarni, S. R., Graham, M. J., et al. 2019, PASP, 131, 018002
Bertin, E. 2009, Mem. Soc. Astron. It., 80, 422
Bertin, E. 2013, Astrophysics Source Code Library [record ascl:1301.001]
Bertin, E., & Arnouts, S. 1996, A&AS, 117, 393
Bosch, J., Armstrong, R., Bickerton, S., et al. 2018, PASJ, 70, S5
Boulade, O., Charlot, X., Abbon, P., et al. 2003, in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, eds. M. Iye, & A. F. M. Moorwood, 4841, 72
Bouy, H., Bertin, E., Moraux, E., et al. 2013, A&A, 554, A101
Casali, M., Adamson, A., Alves de Oliveira, C., et al. 2007, A&A, 467, 777
Cuillandre, J. C., Luppino, G. A., Starr, B. M., & Isani, S. 2000, in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, eds. M. Iye, & A. F. Moorwood, 4008, 1010
Dalton, G. B., Caldwell, M., Ward, A. K., et al. 2006, Proc. SPIE, 6269, 62690X
Flaugher, B. L., Abbott, T. M. C., Annis, J., et al. 2010, in Ground-based and Airborne Instrumentation for Astronomy III, Proc. SPIE, 7735, 77350D
Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., & Garcia-Rodriguez, J. 2017, ArXiv e-prints [arXiv:1704.06857]
Griffin, M. J., Abergel, A., Abreu, A., et al. 2010, A&A, 518, L3
Hampshire, II, J. B., & Pearlmutter, B. 1991, Connectionist Models (Elsevier), 159
Ienaka, N., Kawara, K., Matsuoka, Y., et al. 2013, ApJ, 767, 80
Ives, D. 1998, IEEE Spectrum, 16, 20
Kawanomoto, S., Komiyama, Y., & Yagi, M. 2016a, in Subaru Users' Meeting FY2016
Kawanomoto, Y., Yagi, M., & Kawanomoto, S. 2016b, in Subaru Users' Meeting FY2016
Kingma, D. P., & Ba, J. 2014, ArXiv e-prints [arXiv:1412.6980]
Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, in Advances in Neural Information Processing Systems, 1097
Kuijken, K., Bender, R., Cappellaro, E., et al. 2002, The Messenger, 110, 15
LeCun, Y., & Bengio, Y. 1995, The Handbook of Brain Theory and Neural Networks (Cambridge: MIT Press), 3361
Long, J., Shelhamer, E., & Darrell, T. 2015, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3431
Long, K. S., Baggett, S. M., & MacKenty, J. W. 2015, Persistence in the WFC3 IR Detector: an Improved Model Incorporating the Effects of Exposure Time, Tech. rep.
Lowe, D. G. 1999, ICCV'99: Proceedings of the International Conference on Computer Vision , 1150
Magnier, E. A., & Cuillandre, J.-C. 2004, PASP, 116, 449
Matthews, B. W. 1975, Biochimica et Biophysica Acta (BBA)-Protein Structure, 405, 442
McCully, C., Crawford, S., Kovacs, G., et al. 2018, https://doi.org/10.5281/zenodo.1482019
Melchior, P., Sheldon, E., Drlica-Wagner, A., et al. 2016, Astron. Comput., 16, 99
Metzger, M. R., Luppino, G. A., & Miyazaki, S. 1995, Bull. Am. Astron. Soc., 27, 1389
Miville-Deschênes, M.-A., Duc, P.-A., Marleau, F., et al. 2016, A&A, 593, A4
Miyazaki, S., Komiyama, Y., Kawanomoto, S., et al. 2018, PASJ, 70, S1
Morganson, E., Gruendl, R. A., Menanteau, F., et al. 2018, PASP, 130, 074501
Nir, G., Zackay, B., & Ofek, E. O. 2018, AJ, 156, 229
Ordénovic, C., Surace, C., Torrésani, B., & Llébaria, A. 2008, Stat. Methodol., 5, 373
Pilbratt, G. L., Riedinger, J. R., Passvogel, T., et al. 2010, A&A, 518, L1
Rheault, J. P., Mondrik, N. P., DePoy, D. L., Marshall, J. L., & Suntzeff, N. B. 2014, Spectrophotometric Calibration of the Swope and duPont Telescopes for the Carnegie Supernova Project 2
Richard, M. D., & Lippmann, R. P. 1991, Neural Comput., 3, 461
Rojas, R. 1996, Neural Comput., 8, 41
Rubinstein, R. 1999, Methodol. Comput. Appl. Probab., 1, 127
Ruder, S. 2016, ArXiv e-prints [arXiv:1609.04747]
Saerens, M., Latinne, P., & Decaestecker, C. 2002, Neural Comput., 14, 21
Simonyan, K., & Zisserman, A. 2014, ArXiv e-prints [arXiv:1409.1556]
Szegedy, C., Liu, W., Jia, Y., et al. 2015, ArXiv e-prints [arXiv:1409.4842]
Valdes, F., Gruendl, R., & DES Project 2014, in Astronomical Data Analysis Software and Systems XXIII, eds. N. Manset, & P. Forshay, ASP Conf. Ser., 485, 379
van Dokkum, P. G. 2001, PASP, 113, 1420
Vandame, B. 2002, in Astronomical Data Analysis II, eds. J. L. Starck, & F. D. Murtagh, SPIE Conf. Ser., 4847, 123
Williams, C. K. I. 1998, in Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond, ed. M. I. Jordan (Dordrecht: Springer), 599
Wolfe, T., Armandroff, T., Blouke, M. M., et al. 2000, in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, eds. M. M. Blouke, N. Sampat, G. M. Williams, & T. Yeh, 3965, 80
Yang, T., Wu, Y., Zhao, J., & Guan, L. 2018, Cognit. Syst. Res., 53, 20

A&A 634, A48 (2020)

Appendix A: Performance metric curves and qualitative tests

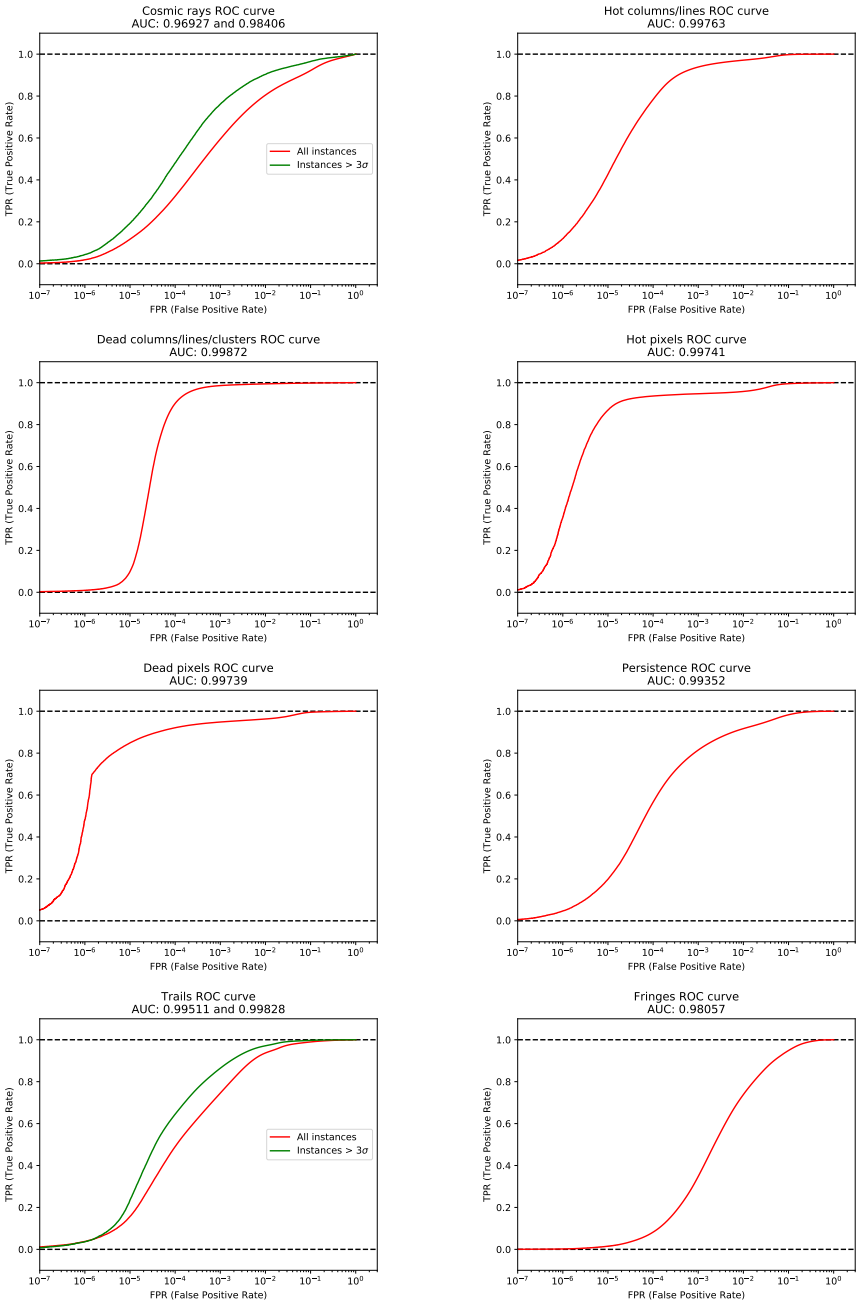


Fig. A.1. ROC curves: TPR vs. FPR. The FPR axis in in logarithmic scale so that very low FPR are best visualized. The ROC curve and the AUC are provided for each class.

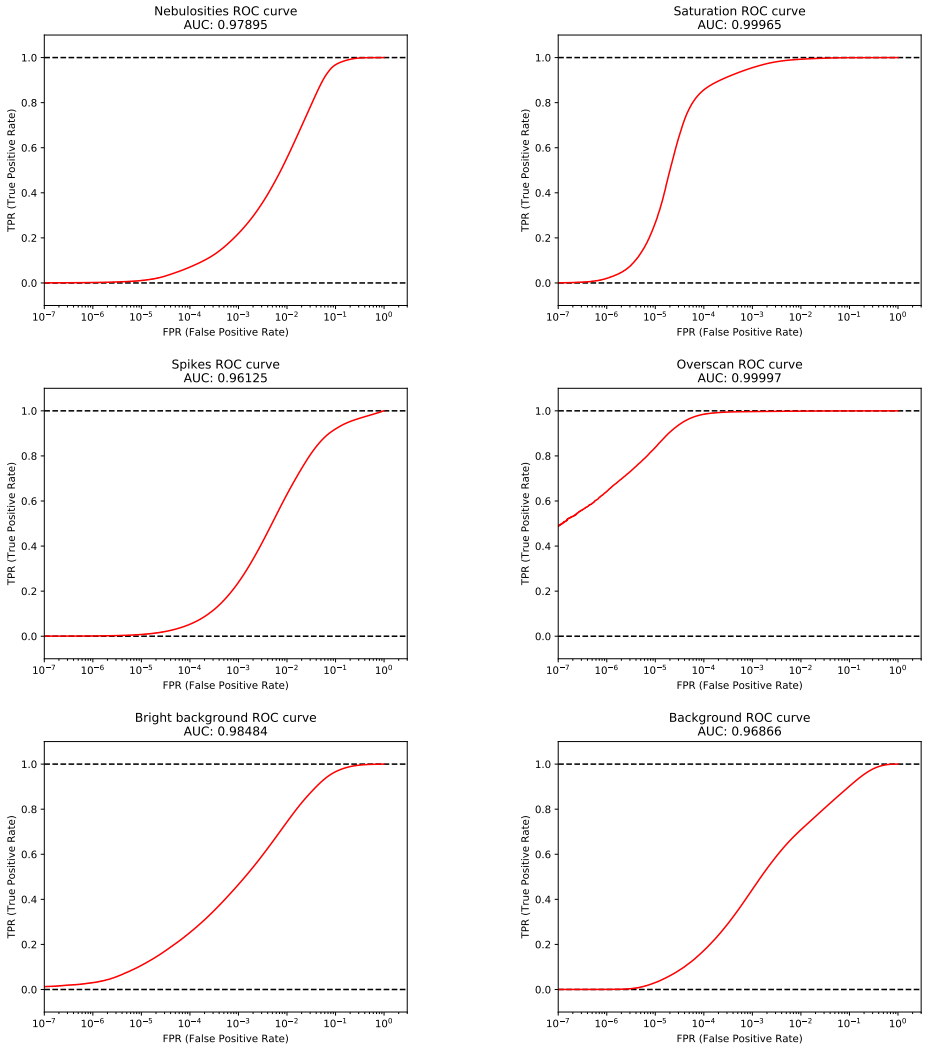


Fig. A.1. continued.

A&A 634, A48 (2020)

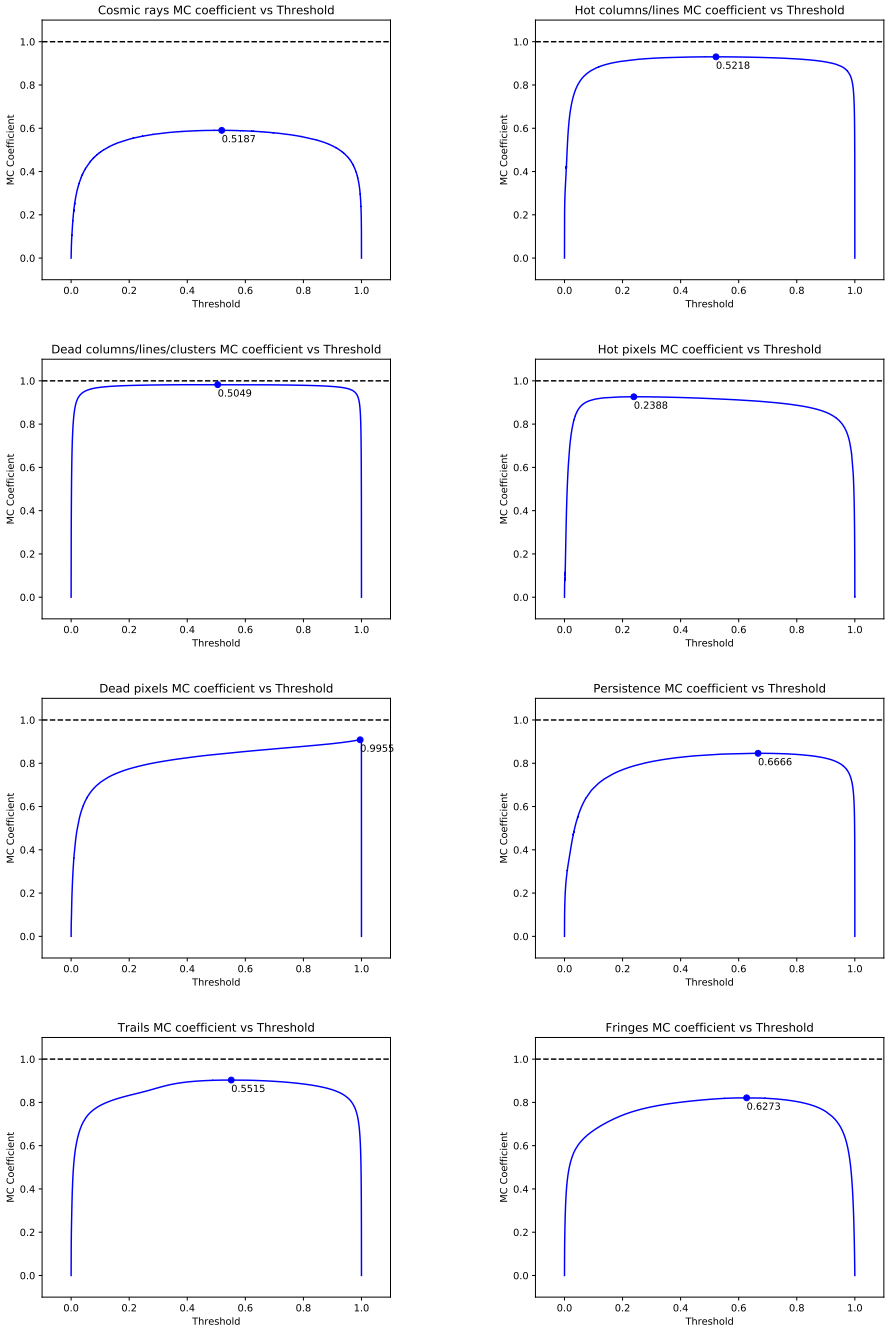


Fig. A.2. MC coefficient curves: MC coefficient vs. detection threshold. On each curve is annotated the threshold for which the MC coefficient is the highest. These curves were computed using the probabilities corrected from priors using empirical training priors.

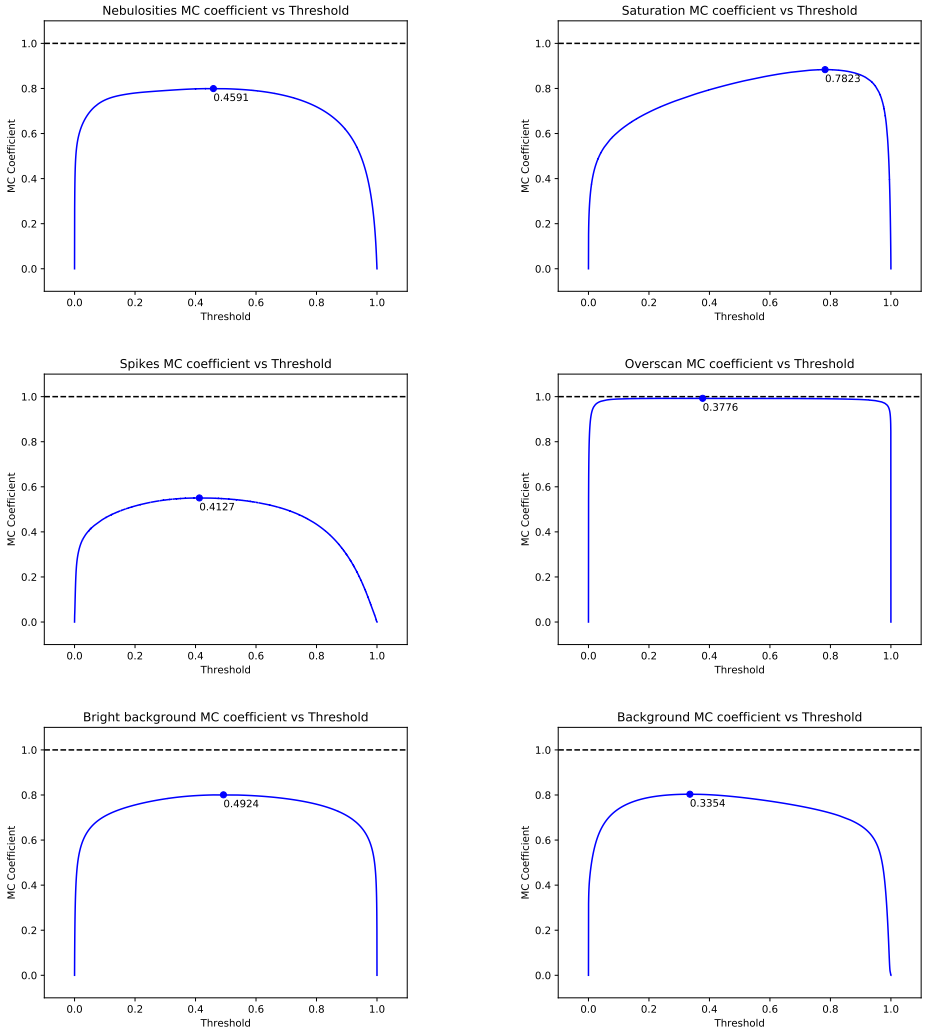


Fig. A.2. continued.

A&A 634, A48 (2020)

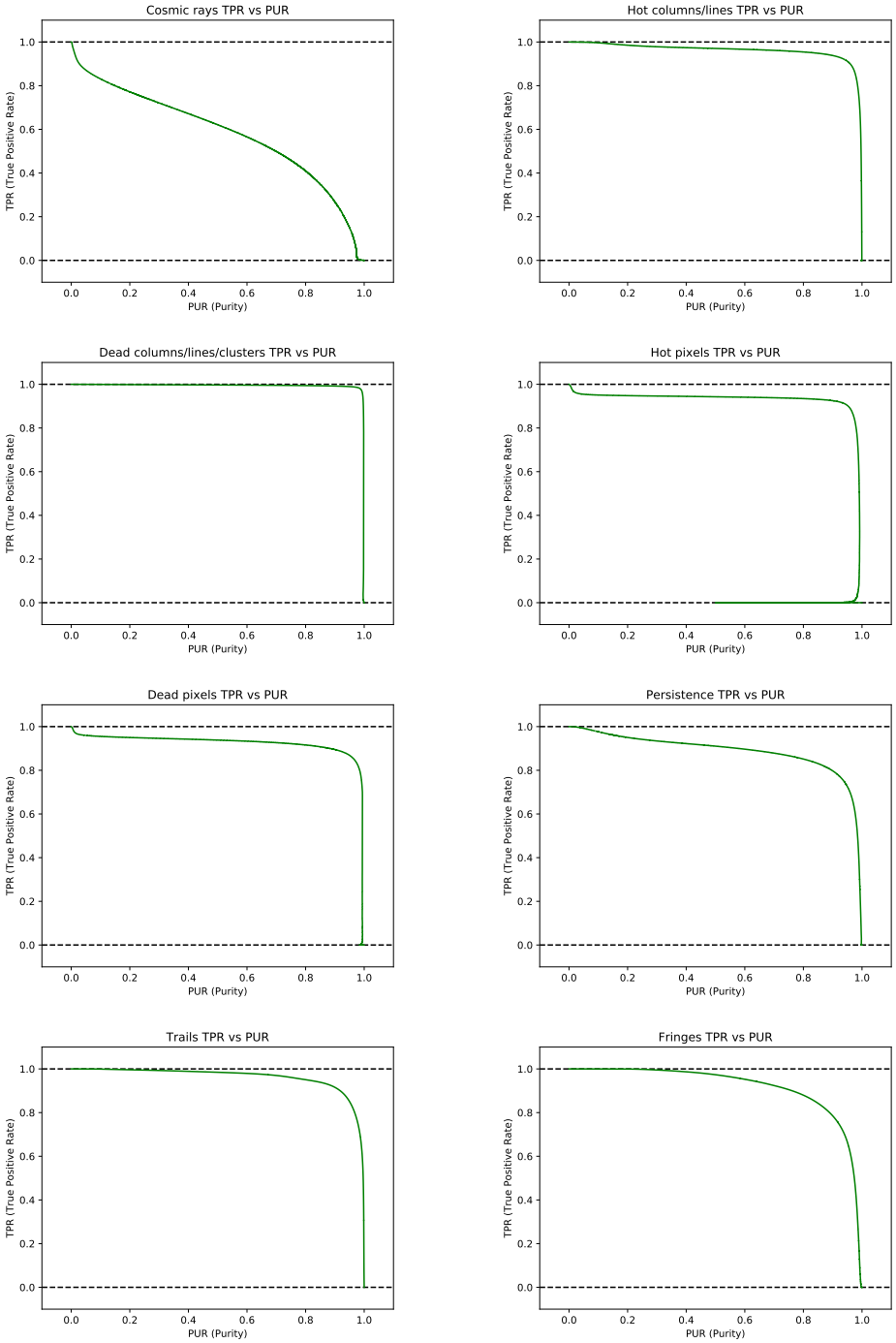


Fig. A.3. Purity curves: TPR vs. PUR.

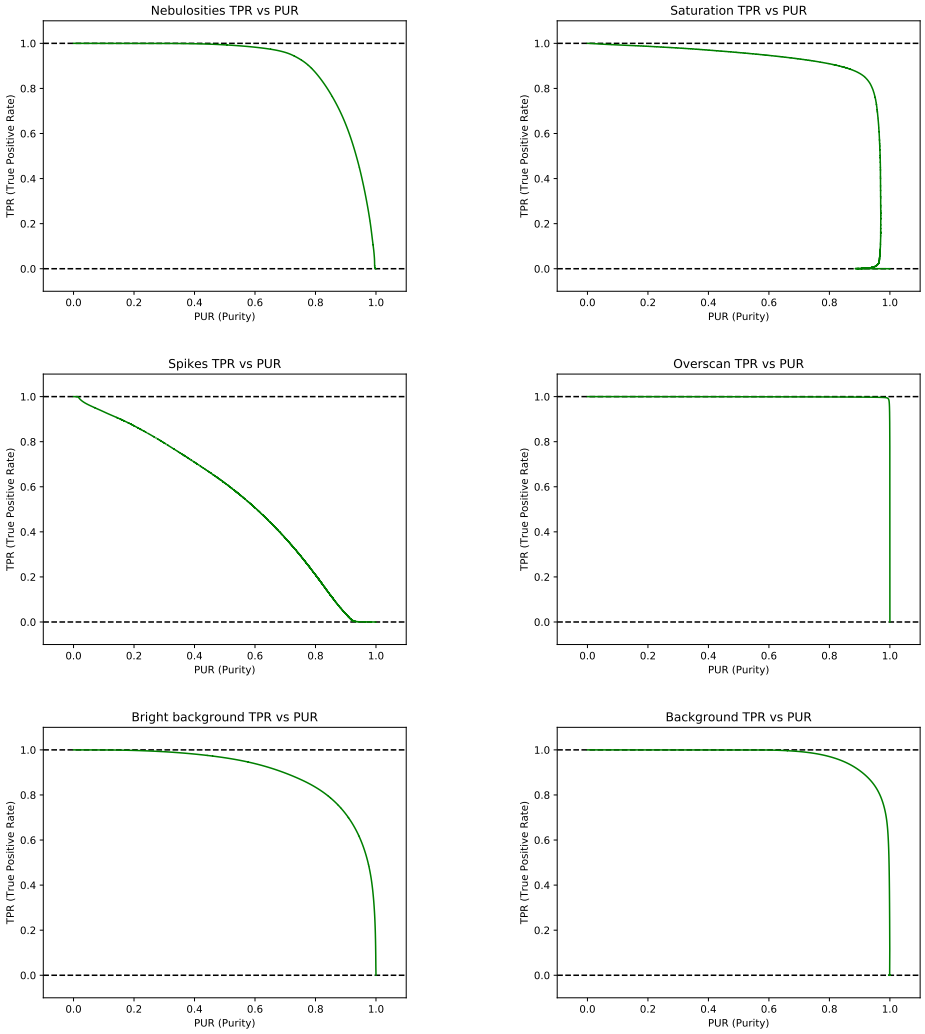


Fig. A.3. continued.

A&A 634, A48 (2020)

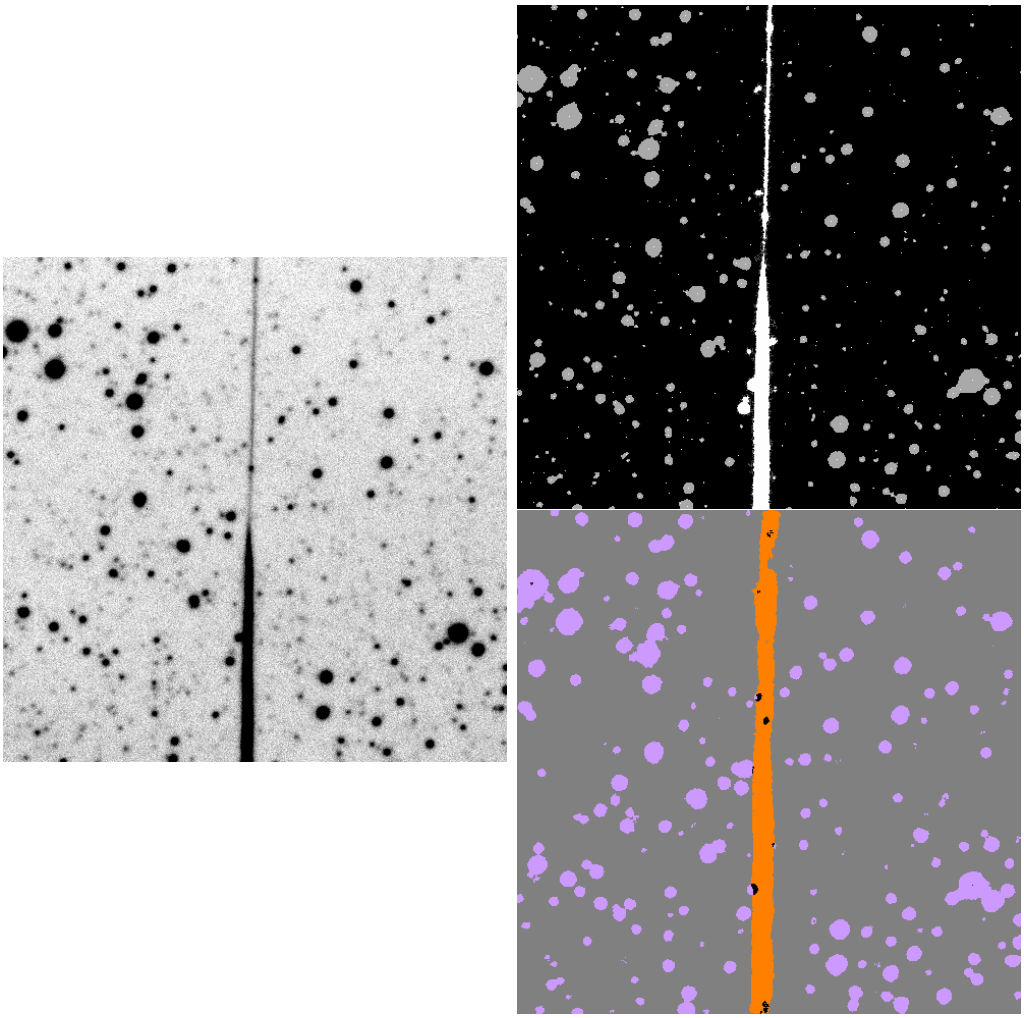


Fig. A.4. Prediction example for an instrument not used in training: ZTF (Bellm et al. 2019). *Left*: a science image exposure. *Top right*: mask from the ZTF pipeline. *Bottom right*: flagging by MaxiMask; the trail is correctly recovered. Also, MaxiMask CNN is able to correctly flag pixels where the trail overlaps sources whereas in the ZTF pipeline, all pixels (i.e., pixels belonging only to the trail, pixels belonging only to sources, and pixels belonging to both the trail and sources) are flagged as both trail and source.

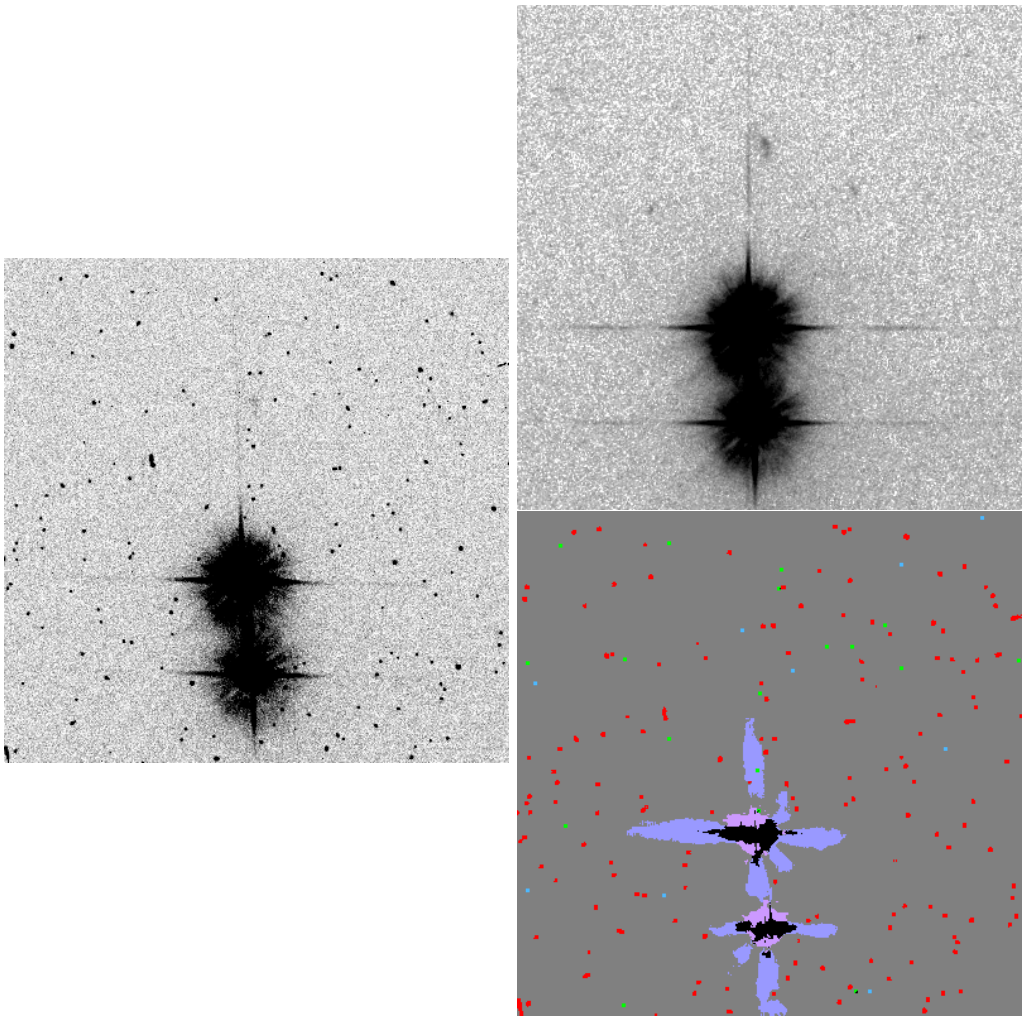


Fig. A.5. Example of a prediction for a space instrument (HST) not used in training (ACS exposure). *Left*: a calibrated (flat-fielded, CTE-corrected) individual exposure of a stellar field in the Pleiades. *Top right*: fully calibrated, geometrically-corrected, dither-combined image where cosmic rays and artifacts have been removed. *Bottom right*: MAXIMASK contaminant identification. Each class is assigned a color so that the ground truth can be represented as a single image (red: CR, dark green: HCL, dark blue: BCL, green: HP, blue: BP, yellow: P, orange: TRL, gray: FR, light gray: NEB, purple: SAT, light purple: SP, brown: OV, pink: BBG, dark gray: BG). Pixels that belong to several classes are represented in black. For the sake of visualization, hot and dead pixel masks have been morphologically dilated so that they appear as 3×3 pixel areas in this representation.

A&A 634, A48 (2020)

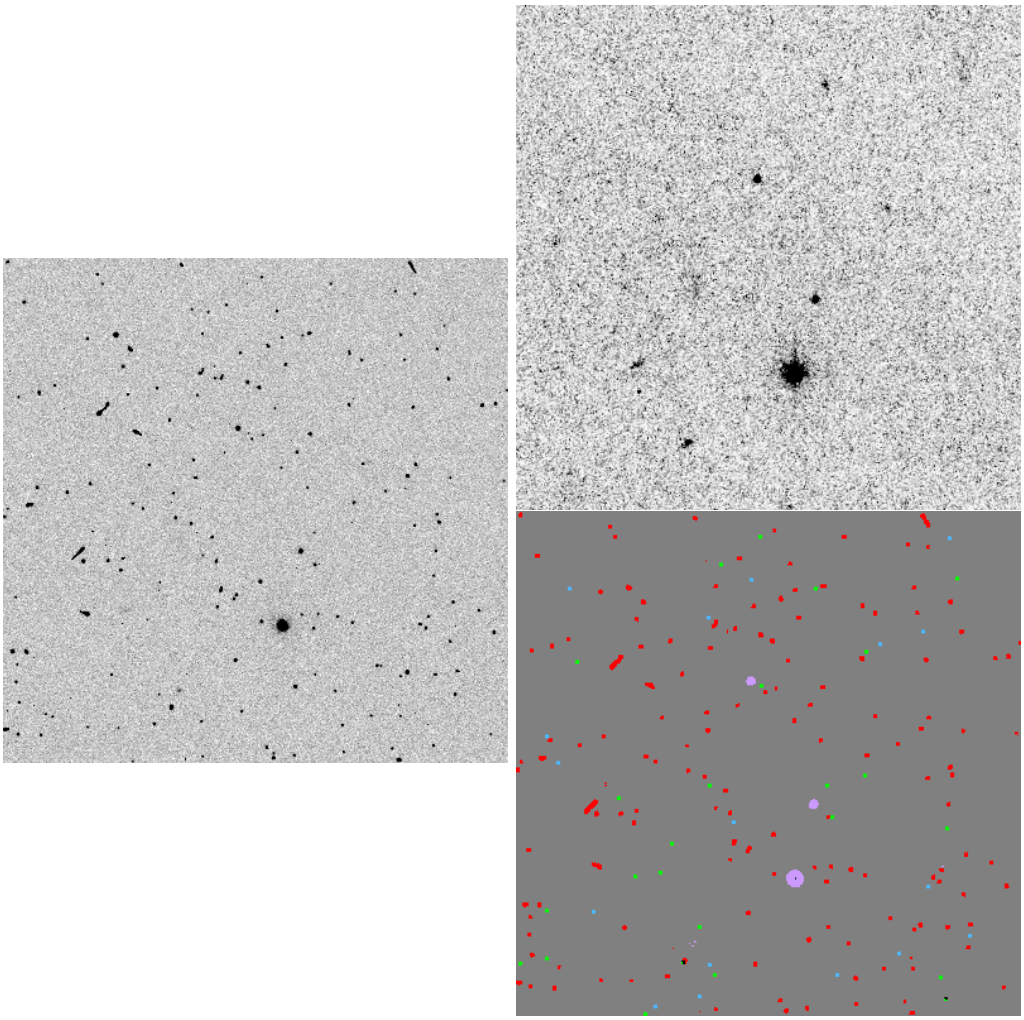


Fig. A.6. Same as Fig. A.5 at a different location in the field to illustrate the ability of MaxiMask to differentiate poorly sampled stellar images from cosmic rays.

List of Figures

Bibliography

M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016. 5.4.2, 5.5.3, 5.5.4, 5.7, 6.5.6

H. Abdi and L. J. Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010. 4.2.7

R. G. Abraham and P. G. van Dokkum. Ultra-Low Surface Brightness Imaging with the Dragonfly Telephoto Array. PASP, 126(935):55, Jan. 2014. doi: 10.1086/674875. 1, C

H. Aihara, N. Arimoto, R. Armstrong, S. Arnouts, N. A. Bahcall, S. Bickerton, J. Bosch, K. Bundy, P. L. Capak, J. H. H. Chan, M. Chiba, J. Coupon, E. Egami, M. Enoki, F. Finet, H. Fujimori, S. Fujimoto, H. Furusawa, J. Furusawa, T. Goto, A. Goulding, J. P. Greco, J. E. Greene, J. E. Gunn, T. Hamana, Y. Harikane, Y. Hashimoto, T. Hattori, M. Hayashi, Y. Hayashi, K. G. Hełminiak, R. Higuchi, C. Hikage, P. T. P. Ho, B.-C. Hsieh, K. Huang, S. Huang, H. Ikeda, M. Imanishi, A. K. Inoue, K. Iwasawa, I. Iwata, A. T. Jaelani, H.-Y. Jian, Y. Kamata, H. Karoji, N. Kashikawa, N. Katayama, S. Kawanomoto, I. Kayo, J. Koda, M. Koike, T. Kojima, Y. Komiyama, A. Konno, S. Koshida, Y. Koyama, H. Kusakabe, A. Leauthaud, C.-H. Lee, L. Lin, Y.-T. Lin, R. H. Lupton, R. Mand elbaum, Y. Matsuoka, E. Medezinski, S. Mineo, S. Miyama, H. Miyatake, S. Miyazaki, R. Momose, A. More, S. More, Y. Moritani, T. J. Moriya, T. Morokuma, S. Mukae, R. Murata, H. Murayama, T. Nagao, F. Nakata, M. Niida, H. Niikura, A. J. Nishizawa, Y. Obuchi, M. Oguri, Y. Oishi, N. Okabe, S. Okamoto, Y. Okura, Y. Ono, M. Onodera, M. Onoue, K. Osato, M. Ouchi, P. A. Price, T.-S. Pyo, M. Sako, M. Sawicki, T. Shibuya, K. Shimasaku, A. Shimono, M. Shirasaki, J. D. Silverman, M. Simet, J. Speagle, D. N. Spergel, M. A. Strauss, Y. Sugahara, N. Sugiyama, Y. Suto, S. H. Suyu, N. Suzuki, P. J. Tait, M. Takada, T. Takata, N. Tamura, M. M. Tanaka, M. Tanaka, M. Tanaka, Y. Tanaka, T. Terai, Y. Terashima, Y. Toba, N. Tominaga, J. Toshikawa, E. L. Turner, T. Uchida, H. Uchiyama, K. Umetsu, F. Uraguchi, Y. Urata, T. Usuda, Y. Utsumi, S.-Y. Wang, W.-H. Wang, K. C. Wong, K. Yabe, Y. Yamada, H. Yamanoi, N. Yasuda, S. Yeh, A. Yonehara, and S. Yuma. The Hyper Suprime-Cam SSP Survey: Overview and survey design. PASJ, 70:S4, Jan. 2018. doi: 10.1093/pasj/psx066. 1, C

G. B. Airy. On the Diffraction of an Object-glass with Circular Aperture. *Transactions of the Cambridge Philosophical Society*, 5:283, Jan. 1835. 2.4.2

S. Alam, F. D. Albareti, C. Allende Prieto, F. Anders, S. F. Anderson, T. Anderton, B. H. Andrews, E. Armengaud, É. Aubourg, S. Bailey, S. Basu, J. E. Bautista, R. L. Beaton, T. C. Beers, C. F. Bender, A. A. Berlind, F. Beutler, V. Bhardwaj, J. C. Bird, D. Bizyaev, C. H. Blake, M. R. Blanton, M. Blomqvist, J. J. Bochanski, A. S. Bolton, J. Bovy, A. Shelden Bradley, W. N. Brandt, D. E. Brauer, J. Brinkmann, P. J. Brown, J. R. Brownstein, A. Burden, E. Burtin, N. G. Busca, Z. Cai, D. Capozzi, A. Carnero Rosell, M. A. Carr, R. Carrera, K. C. Chambers, W. J. Chaplin, Y.-C. Chen, C. Chiappini, S. D. Chojnowski, C.-H. Chuang,

N. Clerc, J. Comparat, K. Covey, R. A. C. Croft, A. J. Cuesta, K. Cunha, L. N. da Costa, N. Da Rio, J. R. A. Davenport, K. S. Dawson, N. De Lee, T. Delubac, R. Deshpande, S. Dhital, L. Dutra-Ferreira, T. Dwelly, A. Ealet, G. L. Ebelke, E. M. Edmondson, D. J. Eisenstein, T. Ellsworth, Y. Elsworth, C. R. Epstein, M. Eracleous, S. Escoffier, M. Esposito, M. L. Evans, X. Fan, E. Fernández-Alvar, D. Feuillet, N. Filiz Ak, H. Finley, A. Finoguenov, K. Flaherty, S. W. Fleming, A. Font-Ribera, J. Foster, P. M. Frinchaboy, J. G. Galbraith-Frew, R. A. García, D. A. García-Hernández, A. E. García Pérez, P. Gaulme, J. Ge, R. Génova-Santos, A. Georgakakis, L. Ghezzi, B. A. Gillespie, L. Girardi, D. Goddard, S. G. A. Gontcho, J. I. González Hernández, E. K. Grebel, P. J. Green, J. N. Grieb, N. Grieves, J. E. Gunn, H. Guo, P. Harding, S. Hasselquist, S. L. Hawley, M. Hayden, F. R. Hearty, S. Hekker, S. Ho, D. W. Hogg, K. Holley-Bockelmann, J. A. Holtzman, K. Honscheid, D. Huber, J. Huehnerhoff, I. I. Ivans, L. Jiang, J. A. Johnson, K. Kinemuchi, D. Kirkby, F. Kitaura, M. A. Klaene, G. R. Knapp, J.-P. Kneib, X. P. Koenig, C. R. Lam, T.-W. Lan, D. Lang, P. Laurent, J.-M. Le Goff, A. Leauthaud, K.-G. Lee, Y. S. Lee, T. C. Licquia, J. Liu, D. C. Long, M. López-Corredoira, D. Lorenzo-Oliveira, S. Lucatello, B. Lundgren, R. H. Lupton, I. Mack, Claude E., S. Mahadevan, M. A. G. Maia, S. R. Majewski, E. Malanushenko, V. Malanushenko, A. Manchado, M. Manera, Q. Mao, C. Maraston, R. C. Marchwinski, D. Margala, S. L. Martell, M. Martig, K. L. Masters, S. Mathur, C. K. McBride, P. M. McGehee, I. D. McGreer, R. G. McMahon, B. Ménard, M.-L. Menzel, A. Merloni, S. Mészáros, A. A. Miller, J. Miralda-Escudé, H. Miyatake, A. D. Montero-Dorta, S. More, E. Morganson, X. Morice-Atkinson, H. L. Morrison, B. Mosser, D. Muna, A. D. Myers, K. Nand ra, J. A. Newman, M. Neyrinck, D. C. Nguyen, R. C. Nichol, D. L. Nidever, P. Noterdaeme, S. E. Nuza, J. E. O'Connell, R. W. O'Connell, R. O'Connell, R. L. C. Ogando, M. D. Olmstead, A. E. Oravetz, D. J. Oravetz, K. Osumi, R. Owen, D. L. Padgett, N. Padmanabhan, M. Paegert, N. Palanque-Delabrouille, K. Pan, J. K. Parejko, I. Pâris, C. Park, P. Pattarakijwanich, M. Pellejero-Ibanez, J. Pepper, W. J. Percival, I. Pérez-Fournon, I. Prez-Ra'fols, P. Petitjean, M. M. Pieri, M. H. Pinsonneault, G. F. Porto de Mello, F. Prada, A. Prakash, A. M. Price-Whelan, P. Protopapas, M. J. Raddick, M. Rahman, B. A. Reid, J. Rich, H.-W. Rix, A. C. Robin, C. M. Rockosi, T. S. Rodrigues, S. Rodríguez-Torres, N. A. Roe, A. J. Ross, N. P. Ross, G. Rossi, J. J. Ruan, J. A. Rubiño-Martín, E. S. Rykoff, S. Salazar-Albornoz, M. Salvato, L. Samushia, A. G. Sánchez, B. Santiago, C. Sayres, R. P. Schiavon, D. J. Schlegel, S. J. Schmidt, D. P. Schneider, M. Schultheis, A. D. Schwope, C. G. Scóccola, C. Scott, K. Sellgren, H.-J. Seo, A. Serenelli, N. Shane, Y. Shen, M. Shetrone, Y. Shu, V. Silva Aguirre, T. Sivarani, M. F. Skrutskie, A. Slosar, V. V. Smith, F. Sobreira, D. Souto, K. G. Stassun, M. Steinmetz, D. Stello, M. A. Strauss, A. Streblyanska, N. Suzuki, M. E. C. Swanson, J. C. Tan, J. Tayar, R. C. Terrien, A. R. Thakar, D. Thomas, N. Thomas, B. A. Thompson, J. L. Tinker, R. Tojeiro, N. W. Troup, M. Vargas-Magaña, J. A. Vazquez, L. Verde, M. Viel, N. P. Vogt, D. A. Wake, J. Wang, B. A. Weaver, D. H. Weinberg, B. J. Weiner, M. White, J. C. Wilson, J. P. Wisniewski, W. M. Wood-Vasey, C. Ye'che, D. G. York, N. L. Zakamska, O. Zamora, G. Zasowski, I. Zehavi, G.-B. Zhao, Z. Zheng, X. Zhou, Z. Zhou, H. Zou, and G. Zhu. The Eleventh and Twelfth Data Releases of the Sloan Digital Sky Survey: Final Data from SDSS-III. ApJS, 219(1):12, July 2015. doi: 10.1088/0067-0049/219/1/12. 1.1, C.1

J. Amiaux, J. L. Auguères, O. Boulade, C. Cara, S. Paulin-Henriksson, A. Réfrégier, S. Ronayette, A. Amara, A. Glauser, C. Dumesnil, A. M. di Giorgio, J. Booth, M. Schweitzer, R. Holmes, M. Cropper, E. Atad-Ettedgui, L. Duvet, and D. Lumb. Euclid imaging channels: from science to system requirements. In *Space Telescopes and Instrumentation 2010: Optical, Infrared, and Millimeter Wave*, volume 7731 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, page 77311I, July 2010. doi: 10.1117/12.857030. 1, C

S. Ando and C. Y. Huang. Deep over-sampling framework for classifying imbalanced data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 770–785. Springer, 2017. 5.5.2

S. Andreon, G. Gargiulo, G. Longo, R. Tagliaferri, and N. Capuano. Wide field imaging - I. Applications of neural networks to object detection and star/galaxy classification. MNRAS, 319(3):700–716, Dec. 2000. doi: 10.1046/j.1365-8711.2000.03700.x. 4.2.7

A. Antoniou, A. Storkey, and H. Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017. 4.2.5

E. Aptoula, S. Lefèvre, and C. Collet. Mathematical morphology applied to the segmentation and classification of galaxies in multispectral images. In *2006 14th European Signal Processing Conference*, pages 1–5. IEEE, 2006. 3.2.2

B. Arcelin, C. Doux, E. Aubourg, and C. Roucelle. Deblending galaxies with Variational Autoencoders: a joint multi-band, multi-instrument approach. *arXiv e-prints*, art. arXiv:2005.12039, May 2020. 6.3

Astropy Collaboration, T. P. Robitaille, E. J. Tollerud, P. Greenfield, M. Droettboom, E. Bray, T. Aldcroft, M. Davis, A. Ginsburg, A. M. Price-Whelan, W. E. Kerzendorf, A. Conley, N. Crighton, K. Barbary, D. Muna, H. Ferguson, F. Grollier, M. M. Parikh, P. H. Nair, H. M. Unther, C. Deil, J. Woillez, S. Conseil, R. Kramer, J. E. H. Turner, L. Singer, R. Fox, B. A. Weaver, V. Zabalza, Z. I. Edwards, K. Azalee Bostroem, D. J. Burke, A. R. Casey, S. M. Crawford, N. Dencheva, J. Ely, T. Jenness, K. Labrie, P. L. Lim, F. Pierfederici, A. Pontzen, A. Ptak, B. Refsdal, M. Servillat, and O. Streicher. Astropy: A community Python package for astronomy. A&A, 558:A33, Oct. 2013. doi: 10.1051/0004-6361/201322068. 5.7

R. G. Autry, R. G. Probst, B. M. Starr, K. M. Abdel-Gawad, R. D. Blakley, P. N. Daly, R. Dominguez, E. A. Hileman, M. Liang, E. T. Pearson, R. A. Shaw, and D. Tody. NEWFIRM: the widefield IR imager for NOAO 4-m telescopes. In M. Iye and A. F. M. Moorwood, editors, Proc. SPIE, volume 4841 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 525–539, Mar. 2003. doi: 10.1117/12.460419. 5.1

V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015. 5.3.1, 5.19, 5.4.2, 5.5.1

V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 1, 5.3.1, 5.19, 5.5.2, 6.4.4, 7, C, D, E

C. A. Bailer-Jones, K. Smith, C. Tiede, R. Sordo, and A. Vallenari. Finding rare objects and building pure samples: probabilistic quasar classification from low-resolution gaia spectra. *Monthly Notices of the Royal Astronomical Society*, 391(4):1838–1853, 2008. 5.5.5

A. Baillard, C. Berger, E. Bertin, T. Géraud, R. Levillain, and N. Widynski. Algorithme de calcul de l'arbre des composantes avec applications à la reconnaissance des formes en imagerie satellitaire. In *Proceedings of the 21st Symposium on Signal and Image Processing (GRETSI)*, Troyes, France, Sept. 2007. 3.2.2

A. Baillard, E. Bertin, V. de Lapparent, P. Fouqué, S. Arnouts, Y. Mellier, R. Pelló, J. F. Leborgne, P. Prugniel, D. Makarov, L. Makarova, H. J. McCracken, A. Bijaoui, and L. Tasca. The EFIGI catalogue of 4458 nearby galaxies with detailed morphology. A&A, 532:A74, Aug 2011. doi: 10.1051/0004-6361/201016423. 2.7, 6.5.1

H. Baird. Document image analysis. chapter document image defect models. *IEEE Computer Society Press, Los Alamitos, CA, USA*, 2:315–325, 1995. 4.2.5

E. B. Baum and F. Wilczek. Supervised learning of probability distributions by neural networks. In *Neural information processing systems*, pages 52–61, 1988. 4.2.4

J. Baxter. Learning internal representations. In *Proceedings of the eighth annual conference on Computational learning theory*, pages 311–320, 1995. 4.2.5

S. Beckouche, J. L. Starck, and J. Fadili. Astronomical image denoising using dictionary learning. A&A, 556:A132, Aug. 2013. doi: 10.1051/0004-6361/201220752. 3.3.3, D

D. Bektešević and D. Vinković. Linear feature detection algorithm for astronomical surveys - I. Algorithm description. MNRAS, 471(3):2626–2641, Nov. 2017. doi: 10.1093/mnras/stx1565. 5.2, D

C. Berger, T. Géraud, R. Levillain, N. Widynski, A. Baillard, and E. Bertin. Effective component tree computation with application to pattern recognition in astronomical imaging. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 41–44, San Antonio, TX, USA, September 2007. IEEE. ISBN 978-1-4244-1437-6. 3.2.2

E. Bertin. Mining Pixels: The Extraction and Classification of Astronomical Sources. In A. J. Banday, S. Zaroubi, and M. Bartelmann, editors, *Mining the Sky*, page 353, Jan 2001. doi: 10.1007/10849171_44. 1, 3.11

E. Bertin. SkyMaker: astronomical image simulations made easy. Mem. Soc. Astron. Italiana, 80:422, Jan. 2009. 2.4.2, 2.3, 5.3, 5.4.2, 6.5.1, D, D

E. Bertin. Automated Morphometry with SExtractor and PSFEx. In I. N. Evans, A. Accomazzi, D. J. Mink, and A. H. Rots, editors, *Astronomical Data Analysis Software and Systems XX*, volume 442 of *Astronomical Society of the Pacific Conference Series*, page 435, July 2011. 5.4.2, 5.4.2

E. Bertin and S. Arnouts. SExtractor: Software for source extraction. A&AS, 117:393–404, Jun 1996. doi: 10.1051/aas:1996164. 1, 3.1.1, 3.1.2, 3.1.5, 3.1.6, 3.8, 4.2.7, 5.4.1, 6.6.1, C, D

E. Bertin, C. Marmo, and H. Bouy. VisiOmatic 2: a Web Client for Remote Visualization With Real-time Mixing of Multispectral Data. In M. Molinaro, K. Shortridge, and F. Pasian, editors, *Astronomical Data Analysis Software and Systems XXVI*, volume 521 of *Astronomical Society of the Pacific Conference Series*, page 651, Oct. 2019a. 1.1, 1.2, 1.3, C.1, C.2, C.3

E. Bertin, M. Schefer, N. Apostolakos, A. Álvarez-Ayllón, P. Dubath, and M. Kümmel. The SourceXtractor++ software. In *Astronomical Data Analysis Software and Systems XXIX. ASP Conference Series*, in press, 2019b. 7, E

S. Beucher and C. Lantuejoul. International workshop on image processing: Real-time edge and motion detection/estimation, 1979. 3.2.2

S. Beucher and F. Meyer. The morphological approach to segmentation: the watershed transformation. *Mathematical morphology in image processing*, 34:433–481, 1993. 3.2.2

A. Bijaoui. Sky background estimation and application. A&A, 84(1-2):81–84, Apr. 1980. 3.1.1

A. Bijaoui and F. Rué. A multiscale vision model adapted to the astronomical images. *Signal processing*, 46(3):345–362, 1995. 3.3.2, D

M. Bílek, P.-A. Duc, J.-C. Cuilland re, S. Gwyn, M. Cappellari, D. V. Bekaert, P. Bonfini, T. Bitsakis, S. Paudel, D. Krajnović, P. R. Durrell, and F. Marleau. Census and classification of low-surface-brightness structures in nearby early-type galaxies from the MATLAS survey. MNRAS, Aug. 2020. doi: 10.1093/mnras/staa2248. 1, C

C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006. 1, 4.2.5

C. M. Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995. 4.2.4, 4.2.4, 4.2.4, 4.2.4, 4.2.5, 4.2.5

J. R. Blum et al. Approximation methods which converge with probability one. *The Annals of Mathematical Statistics*, 25(2):382–386, 1954. 4.2.1

J. Bobin, J.-L. Starck, and R. Ottensamer. Compressed Sensing in Astronomy. *IEEE Journal of Selected Topics in Signal Processing*, 2(5):718–726, Nov. 2008. doi: 10.1109/JSTSP.2008.2005337. 3.3.3, D

A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 6.2.2

J. Bosch, R. Armstrong, S. Bickerton, H. Furusawa, H. Ikeda, M. Koike, R. Lupton, S. Mineo, P. Price, T. Takata, M. Tanaka, N. Yasuda, Y. AlSayyad, A. C. Becker, W. Coulton, J. Coupon, J. Garmilla, S. Huang, K. S. Krughoff, D. Lang, A. Leauthaud, K.-T. Lim, N. B. Lust, L. A. MacArthur, R. Mandelbaum, H. Miyatake, S. Miyazaki, R. Murata, S. More, Y. Okura, R. Owen, J. D. Swinbank, M. A. Strauss, Y. Yamada, and H. Yamanoi. The Hyper Suprime-Cam software pipeline. PASJ, 70:S5, Jan. 2018. doi: 10.1093/pasj/psx080. 5.2, 5.4.1, D

J. Bosch, Y. AlSayyad, R. Armstrong, E. Bellm, H.-F. Chiang, S. Eggl, K. Findeisen, M. Fisher-Levine, L. P. Guy, A. Guyonnet, Ž. Ivezić, T. Jenness, G. Kovács, K. S. Krughoff, R. H. Lupton, N. B. Lust, L. A. MacArthur, J. Meyers, F. Moolekamp, C. B. Morrison, T. D. Morton, W. O'Mullane, J. K. Parejko, A. A. Plazas, P. A. Price, M. L. Rawls, S. L. Reed, P. Schellart, C. T. Slater, I. Sullivan, J. D. Swinbank, D. Taranu, C. Z. Waters, and W. M. Wood-Vasey. An Overview of the LSST Image Processing Pipelines. In P. J. Teuben, M. W. Pound, B. A. Thomas, and E. M. Warner, editors, *Astronomical Data Analysis Software and Systems XXVII*, volume 523 of *Astronomical Society of the Pacific Conference Series*, page 521, Oct. 2019. 5.2, D

L. Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):142, 1998. 4.2.1

A. Boucaud, M. Huertas-Company, C. Heneka, E. E. O. Ishida, N. Sedaghat, R. S. de Souza, B. Moews, H. Dole, M. Castellano, E. Merlin, V. Roscani, A. Tramacere, M. Killedar, A. M. M. Trindade, and Collaboration COIN. Photometry of high-redshift blended galaxies using deep learning. MNRAS, 491(2):2481–2495, Jan. 2020. doi: 10.1093/mnras/stz3056. 6.3, 6.5.1

O. Boulade, X. Charlot, P. Abbon, S. Aune, P. Borgeaud, P.-H. Carton, M. Carty, J. Da Costa, H. Deschamps, D. Desforge, D. Eppellé, P. Gallais, L. Gosset, R. Granelli, M. Gros, J. de Kat, D. Loiseau, J. . Ritou, J. Y. Roussé, P. Starzynski, N. Vignal, and L. G. Vigroux. MegaCam: the new Canada-France-Hawaii Telescope wide-field imaging camera. In M. Iye and A. F. M. Moorwood, editors, Proc. SPIE, volume 4841 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 72–81, Mar. 2003. doi: 10.1117/12.459890. 5.1

H. Bouy, E. Bertin, E. Moraux, J. C. Cuillandre, J. Bouvier, D. Barrado, E. Solano, and A. Bayo. Dynamical analysis of nearby clusters. Automated astrometry from the ground: precision proper motions over a wide field. A&A, 554:A101, June 2013. doi: 10.1051/0004-6361/201220748. 5.2, 5.4.1, D

H. Bouy, E. Bertin, L. M. Sarro, D. Barrado, A. Berihuete, J. Olivares, E. Moraux, J. Bouvier, M. Tamura, J. C. Cuillandre, Y. Beletsky, N. Wright, N. Huelamo, L. Allen, E. Solano, and B. Brand ner. The COSMIC-DANCE project: Unravelling the origin of the mass function. In S. Arribas, A. Alonso-Herrero, F. Figueras, C. Hernández-Monteagudo, A. Sánchez-Lavega, and S. Pérez-Hoyos, editors, *Highlights on Spanish Astrophysics IX*, pages 338–344, Mar. 2017. 1, C

C. Bowles, L. Chen, R. Guerrero, P. Bentley, R. Gunn, A. Hammers, D. A. Dickie, M. V. Hernández, J. Wardlaw, and D. Rueckert. Gan augmentation: Augmenting training data using generative adversarial networks. *arXiv preprint arXiv:1810.10863*, 2018. 4.2.5

L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996. 4.2.5

B. J. Brewer, D. Foreman-Mackey, and D. W. Hogg. Probabilistic Catalogs for Crowded Stellar Fields. AJ, 146(1):7, July 2013. doi: 10.1088/0004-6256/146/1/7. 3.3.4

J. S. Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing*, pages 227–236. Springer, 1990. 4.2.3

A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 4.2.5

D. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks, complex systems, vol. 2. 1988a. 2

D. S. Broomhead and D. Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. Technical report, Royal Signals and Radar Establishment Malvern (United Kingdom), 1988b. 2

S. R. Bulo, G. Neuhold, and P. Kontschieder. Loss maxpooling for semantic image segmentation. *CVPR), July*, 7, 2017. 5.5.2

R. Buonanno, C. E. Corsi, G. A. de Biase, and I. Ferraro. A Method for Stellar Photometry in Crowded Fields. In G. Sedmak, M. Capaccioli, and R. J. Allen, editors, *Image Processing in Astronomy*, page 354, Jan. 1979. 3.1.5

R. Buonanno, G. Buscema, C. E. Corsi, I. Ferraro, and G. Iannicola. Automated photographic photometry of stars in globular clusters. A&A, 126:278–282, Oct. 1983a. 3.1.1, 3.1.3, 3.1.5

R. Buonanno, G. Buscema, C. E. Corsi, G. Iannicola, and F. Fusi Pecci. Positions, magnitudes, and colors for stars in the globular cluster M15. A&AS, 51:83–92, Jan. 1983b. 3.1.5

C. J. Burke, P. D. Aleo, Y.-C. Chen, X. Liu, J. R. Peterson, G. H. Sembroski, and J. Y.-Y. Lin. Deblending and classifying astronomical sources with Mask R-CNN deep learning. MNRAS, 490(3):3952–3965, Dec. 2019. doi: 10.1093/mnras/stz2845. 6.3

D. J. Burr. Experiments on neural net recognition of spoken and written text. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1162–1168, 1988. 4.2.7

G. Cabrera-Vives, I. Reyes, F. Förster, P. A. Estévez, and J.-C. Maureira. Deep-hits: Rotation invariant convolutional neural network for transient detection. *The Astrophysical Journal*, 836 (1):97, 2017. 4.2.5

J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 6.2.1

Y. Cao and L. Chen. Convolutional neural networks for galaxy zoo challenge of morphological class probablity regression. 4.2.5

S. Carassou, V. de Lapparent, E. Bertin, and D. Le Borgne. Inferring the photometric and size evolution of galaxies from image simulations. I. Method. A&A, 605:A9, Sept. 2017. doi: 10.1051/0004-6361/201730587. 1, C

R. A. Caruana. Multitask connectionist learning. In *In Proceedings of the 1993 Connectionist Models Summer School*. Citeseer, 1993. 4.2.5

P. Carvalho, G. Rocha, and M. P. Hobson. A fast Bayesian approach to discrete object detection in astronomical data sets - PowellSnakes I. MNRAS, 393(3):681–702, Mar. 2009. doi: 10.1111/j.1365-2966.2008.14016.x. 3.3.4

P. Carvalho, G. Rocha, M. P. Hobson, and A. Lasenby. PowellSnakes II: a fast Bayesian approach to discrete object detection in multi-frequency astronomical data sets. MNRAS, 427(2):1384–1400, Dec. 2012. doi: 10.1111/j.1365-2966.2012.22033.x. 3.3.4

M. Casali, A. Adamson, C. Alves de Oliveira, O. Almaini, K. Burch, T. Chuter, J. Elliot, M. Folger, S. Foucaud, N. Hambly, M. Hastie, D. Henry, P. Hirst, M. Irwin, D. Ives, A. Lawrence, K. Laidlaw, D. Lee, J. Lewis, D. Lunney, S. McLay, D. Montgomery, A. Pickup, M. Read, N. Rees, I. Robson, K. Sekiguchi, A. Vick, S. Warren, and B. Woodward. The UKIRT wide-field camera. A&A, 467:777–784, May 2007. doi: 10.1051/0004-6361:20066514. 5.1

L. Cayón, J. L. Sanz, R. B. Barreiro, E. Martínez-González, P. Vielva, L. Toffolatti, J. Silk, J. M. Diego, and F. Argüeso. Isotropic wavelets: a powerful tool to extract point sources from cosmic microwave background maps. MNRAS, 315(4):757–761, July 2000. doi: 10.1046/j.1365-8711.2000.03462.x. 3.3.2, D

C. Chang, M. Jarvis, B. Jain, S. M. Kahn, D. Kirkby, A. Connolly, S. Krughoff, E. H. Peng, and J. R. Peterson. The effective number density of galaxies for weak lensing measurements in the LSST project. MNRAS, 434(3):2121–2135, Sept. 2013. doi: 10.1093/mnras/stt1156. 1, C

C. Chang, M. T. Busha, R. H. Wechsler, A. Refregier, A. Amara, E. Rykoff, M. R. Becker, C. Bruderer, L. Gamper, B. Leistedt, H. Peiris, T. Abbott, F. B. Abdalla, E. Balbinot, M. Banerji, R. A. Bernstein, E. Bertin, D. Brooks, A. Carnero, S. Desai, L. N. da Costa, C. E. Cunha, T. Eifler, A. E. Evrard, A. Fausti Neto, D. Gerdes, D. Gruen, D. James, K. Kuehn, M. A. G. Maia, M. Makler, R. Ogando, A. Plazas, E. Sanchez, B. Santiago, M. Schubnell, I. Sevilla-Noarbe, C. Smith, M. Soares-Santos, E. Suchyta, M. E. C. Swanson, G. Tarle, and J. Zuntz. Modeling the Transfer Function for the Dark Energy Survey. ApJ, 801(2):73, Mar. 2015. doi: 10.1088/0004-637X/801/2/73. 1, C

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002. 5.5.2

X. Chen, N. Mishra, M. Rohaninejad, and P. Abbeel. Pixelsnail: An improved autoregressive generative model. In *International Conference on Machine Learning*, pages 864–872. PMLR, 2018. 6.3

M. Cheselka. Automatic Detection of Linear Features in Astronomical Images. In D. M. Mehringer, R. L. Plante, and D. A. Roberts, editors, *Astronomical Data Analysis Software and Systems VIII*, volume 172 of *Astronomical Society of the Pacific Conference Series*, page 349, Jan. 1999. 5.2

D. Chicco. Ten quick tips for machine learning in computational biology. *BioData mining*, 10 (1):35, 2017. 5.6.1

D. Chicco and G. Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1):6, 2020. 5.6.1

D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011. 9

D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015. 4.2.3

R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008. 4.3.3

E. Contini, G. De Lucia, Á. Villalobos, and S. Borgani. On the formation and physical properties of the intracluster light in hierarchical galaxy formation models. MNRAS, 437(4):3787–3802, Feb. 2014. doi: 10.1093/mnras/stt2174. 1, C

E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 113–123, 2019. 4.2.5

J. C. Cuillandre and E. Bertin. CFHT Legacy Survey (CFHTLS) : a rich data set. In D. Barret, F. Casoli, G. Lagache, A. Lecavelier, and L. Pagani, editors, *SF2A-2006: Semaine de l'Astrophysique Francaise*, page 265, June 2006. 1.1, C.1

J.-C. Cuillandre, G. A. Luppino, B. M. Starr, and S. Isani. Performance of the CFH12K: a 12K by 8K CCD mosaic camera for the CFHT prime focus. In M. Iye and A. F. Moorwood, editors, Proc. SPIE, volume 4008 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 1010–1021, Aug. 2000. doi: 10.1117/12.395465. 5.1

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989. 4.2.2

G. B. Dalton, M. Caldwell, A. K. Ward, M. S. Whalley, G. Woodhouse, R. L. Edeson, P. Clark, S. M. Beard, A. M. Gallie, S. P. Todd, J. M. D. Strachan, N. N. Bezawada, W. J. Sutherland, and J. P. Emerson. The VISTA infrared camera. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 6269 of Proc. SPIE, page 62690X, June 2006. doi: 10.1117/12.670018. 5.1

F. Damiani, A. Maggio, G. Micela, and S. Sciortino. A Method Based on Wavelet Transforms for Source Detection in Photon-counting Detector Images. I. Theory and General Properties. ApJ, 483(1):350–369, July 1997. doi: 10.1086/304217. 3.1.1, 3.3.2

T. Daylan, S. K. N. Portillo, and D. P. Finkbeiner. Inference of Unresolved Point Sources at High Galactic Latitudes Using Probabilistic Catalogs. ApJ, 839(1):4, Apr. 2017. doi: 10.3847/1538-4357/aa679e. 3.3.4

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 4.3.3

S. Desai, J. J. Mohr, E. Bertin, M. Kümmel, and M. Wetzstein. Detection and removal of artifacts in astronomical images. *Astronomy and Computing*, 16:67–78, July 2016. doi: 10.1016/j.ascom.2016.04.002. 5.2

T. DeVries and G. W. Taylor. Dataset augmentation in feature space. *arXiv preprint arXiv:1702.05538*, 2017. 4.2.5

P. Dimauro, M. Huertas-Company, E. Daddi, P. G. Pérez-González, M. Bernardi, G. Barro, F. Buitrago, F. Caro, A. Cattaneo, H. Dominguez-Sánchez, S. r. M. Faber, B. Häußler, D. D. Kocevski, A. M. Koekemoer, D. C. Koo, C. T. Lee, S. Mei, B. Margalef-Bentabol, J. Primack, A. Rodriguez-Puebla, M. Salvato, F. Shankar, and D. Tuccillo. A catalog of polychromatic bulge-disc decompositions of ∼17.600 galaxies in CANDELS. MNRAS, 478(4):5410–5426, Aug. 2018. doi: 10.1093/mnras/sty1379. 6.5.1

S. P. Driver, J. Liske, N. J. G. Cross, R. De Propris, and P. D. Allen. The Millennium Galaxy Catalogue: the space density and surface-brightness distribution(s) of galaxies. MNRAS, 360 (1):81–103, June 2005. doi: 10.1111/j.1365-2966.2005.08990.x. 1, 1.2, C, C.2

P.-A. Duc. MATLAS: a deep exploration of the surroundings of massive early-type galaxies. *arXiv e-prints*, art. arXiv:2007.13874, July 2020. 1, C

P.-A. Duc, J.-C. Cuillandre, E. Karabal, M. Cappellari, K. Alatalo, L. Blitz, F. Bournaud, M. Bureau, A. F. Crocker, R. L. Davies, T. A. Davis, P. T. de Zeeuw, E. Emsellem, S. Khochfar, D. Krajnović, H. Kuntschner, R. M. McDermid, L. Michel-Dansac, R. Morganti, T. Naab, T. Oosterloo, S. Paudel, M. Sarzi, N. Scott, P. Serra, A.-M. Weijmans, and L. M. Young. The ATLAS^{3D} project - XXIX. The new look of early-type galaxies and surrounding fields disclosed by extremely deep optical images. MNRAS, 446(1):120–143, Jan. 2015. doi: 10.1093/mnras/stu2019. 1.3, C.3

C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, and R. Garcia. Incorporating second-order functional knowledge for better option pricing. In *Advances in neural information processing systems*, pages 472–478, 2001. 4.2.3

V. Dumoulin and F. Visin. A guide to convolution arithmetic for deep learning. *ArXiv e-prints*, mar 2016. 5.3.1, 5.16

T. Erben, M. Schirmer, J. P. Dietrich, O. Cordes, L. Haberzettl, M. Hetterscheidt, H. Hildebrandt, O. Schmithuesen, P. Schneider, P. Simon, E. Deul, R. N. Hook, N. Kaiser, M. Radovich, C. Benoist, M. Nonino, L. F. Olsen, I. Prandoni, R. Wichmann, S. Zaggia, D. Bomans, R. J. Dettmar, and J. M. Miralles. GaBoDS: The Garching-Bonn Deep Survey. IV. Methods for the image reduction of multi-chip cameras demonstrated on data from the ESO Wide-Field Imager. *Astronomische Nachrichten*, 326(6):432–464, July 2005. doi: 10.1002/asna.200510396. 5.2

M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 4.3.3

J. M. Fadili and J.-L. Starck. Curvelets and ridgelets, 2009. 3.3.2

C. L. Farage and K. A. Pimbblet. Evaluation of Cosmic Ray Rejection Algorithms on Single-Shot Exposures. PASA, 22(3):249–256, Aug. 2005. doi: 10.1071/AS05012. 5.2

R. M. Feder, S. K. N. Portillo, T. Daylan, and D. Finkbeiner. Multiband Probabilistic Cataloging: A Joint Fitting Approach to Point-source Detection and Deblending. AJ, 159(4):163, Apr. 2020. doi: 10.3847/1538-3881/ab74cf. 3.3.4

B. L. Flaugher, T. M. C. Abbott, J. Annis, M. L. Antonik, J. Bailey, O. Ballester, J. P. Bernstein, R. Bernstein, M. Bonati, G. Bremer, J. Briones, D. Brooks, E. J. Buckley-Geer, J. Campa, L. Cardiel-Sas, F. Castander, J. Castilla, H. Cease, S. Chappa, E. C. Chi, L. da Costa, D. L. DePoy, G. Derylo, J. De Vicente, H. T. Diehl, P. Doel, J. Estrada, J. Eiting, A. Elliott, D. Finley, J. Frieman, E. Gaztanaga, D. Gerdes, M. Gladders, V. Guarino, G. Gutierrez, J. Grudzinski, B. Hanlon, J. Hao, S. Holland, K. Honscheid, D. Huffman, C. Jackson, I. Karliner, D. Kau, S. Kent, K. Krempetz, J. Krider, M. Kozlovsky, D. Kubik, K. W. Kuehn, S. E. Kuhlmann, K. Kuk, O. Lahav, P. Lewis, H. Lin, W. Lorenzon, S. Marshall, G. Martínez, T. McKay, W. Merritt, M. Meyer, R. Miquel, J. Morgan, P. Moore, T. Moore, B. Nord, R. Ogando, J. Olsen, J. Peoples, A. Plazas, N. Roe, A. Roodman, B. Rossetto, E. Sanchez, V. Scarpine, T. Schalk, R. Schindler, R. Schmidt, R. Schmitt, M. Schubnell, K. Schultz, M. Selen, S. Serrano, T. Shaw, V. Simaitis, J. Slaughter, R. C. Smith, H. Spinka, A. Stefanik, W. Stuermer, A. Sypniewski, R. Talaga, G. Tarle, J. Thaler, D. Tucker, A. R. Walker, C. Weaverdyck, W. Wester, R. J. Woods, S. Worswick, and A. Zhao. Status of the dark energy survey camera (DECam) project. In *Ground-based and Airborne Instrumentation for Astronomy III*, volume 7735 of Proc. SPIE, page 77350D, July 2010. doi: 10.1117/12.856609. 5.1

H. Flewelling. Pan-STARRS Data Release 1. In *American Astronomical Society Meeting Abstracts #229*, volume 229 of *American Astronomical Society Meeting Abstracts*, page 237.07, Jan. 2017. 1.1, C.1

H. Flewelling. The Pan-STARRS pipeline and data products. In *American Astronomical Society Meeting Abstracts #231*, volume 231 of *American Astronomical Society Meeting Abstracts*, page 102.02, Jan. 2018. 1.1, C.1

O. G. Franz. Observational Procedures for Visual Double-Star Work. JRASC, 67:81, Apr. 1973. 3.1.5

R. Fraser and E. Suzuki. Resolution of overlapping absorption bands by least squares procedures. *Analytical Chemistry*, 38(12):1770–1773, 1966. 3.1.5

P. E. Freeman, V. Kashyap, R. Rosner, and D. Q. Lamb. A Wavelet-Based Algorithm for the Spatial Analysis of Poisson Data. ApJS, 138(1):185–218, Jan. 2002. doi: 10.1086/324017. 3.3.2

A. S. Fruchter and R. N. Hook. Drizzle: A Method for the Linear Reconstruction of Undersampled Images. PASP, 114(792):144–152, Feb. 2002. doi: 10.1086/338393. 5.2

Fu Jie Huang and Y. LeCun. Large-scale learning with svm and convolutional for generic object categorization. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 284–291, 2006. 4.3.2

K. Fukunaga. Introduction to statistical pattern recognition, ser. *Computer science and scientific computing. Boston: Academic Press*, 1990. 3.1.4

K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980. 7

Gaia Collaboration, T. Prusti, J. H. J. de Bruijne, A. G. A. Brown, A. Vallenari, C. Babusiaux, C. A. L. Bailer-Jones, U. Bastian, M. Biermann, D. W. Evans, L. Eyer, F. Jansen, C. Jordi, S. A. Klioner, U. Lammers, L. Lindegren, X. Luri, F. Mignard, D. J. Milligan, C. Panem, V. Poinsignon, D. Pourbaix, S. Randich, G. Sarri, P. Sartoretti, H. I. Siddiqui, C. Soubiran, V. Valette, F. van Leeuwen, N. A. Walton, C. Aerts, F. Arenou, M. Cropper, R. Drimmel, E. Høg, D. Katz, M. G. Lattanzi, W. O'Mullane, E. K. Grebel, A. D. Holland, C. Huc, X. Passot, L. Bramante, C. Cacciari, J. Castañeda, L. Chaoul, N. Cheek, F. De Angeli, C. Fabricius, R. Guerra, J. Hernández, A. Jean-Antoine-Piccolo, E. Masana, R. Messineo, N. Mowlavi, K. Nienartowicz, D. Ordóñez-Blanco, P. Panuzzo, J. Portell, P. J. Richards, M. Riello, G. M. Seabroke, P. Tanga, F. Thévenin, J. Torra, S. G. Els, G. Gracia-Abril, G. Comoretto, M. Garcia-Reinaldos, T. Lock, E. Mercier, M. Altmann, R. Andrae, T. L. Astraatmadja, I. Bellas-Velidis, K. Benson, J. Berthier, R. Blomme, G. Busso, B. Carry, A. Cellino, G. Clementini, S. Cowell, O. Creevey, J. Cuypers, M. Davidson, J. De Ridder, A. de Torres, L. Delchambre, A. Dell'Oro, C. Ducourant, Y. Frémat, M. García-Torres, E. Gosset, J. L. Halbwachs, N. C. Hambly, D. L. Harrison, M. Hauser, D. Hestroffer, S. T. Hodgkin, H. E. Huckle, A. Hutton, G. Jasniewicz, S. Jordan, M. Kontizas, A. J. Korn, A. C. Lanzafame, M. Manteiga, A. Moitinho, K. Muinonen, J. Osinde, E. Pancino, T. Pauwels, J. M. Petit, A. Recio-Blanco, A. C. Robin, L. M. Sarro, C. Siopis, M. Smith, K. W. Smith, A. Sozzetti, W. Thuillot, W. van Reeven, Y. Viala, U. Abbas, A. Abreu Aramburu, S. Accart, J. J. Aguado, P. M. Allan, W. Allasia, G. Altavilla, M. A. Álvarez, J. Alves, R. I. Anderson, A. H. Andrei, E. Anglada Varela, E. Antiche, T. Antoja, S. Antón, B. Arcay, A. Atzei, L. Ayache, N. Bach, S. G. Baker, L. Balaguer-Núñez, C. Barache, C. Barata, A. Barbier, F. Barblan, M. Baroni, D. Barrado y Navascués, M. Barros, M. A. Barstow, U. Becciani, M. Bellazzini, G. Bellei, A. Bello García, V. Belokurov, P. Bendjoya, A. Berihuete, L. Bianchi, O. Bienaymé, F. Billebaud, N. Blagorodnova, S. Blanco-Cuaresma, T. Boch, A. Bombrun, R. Borrachero, S. Bouquillon, G. Bourda, H. Bouy, A. Bragaglia, M. A. Breddels, N. Brouillet, T. Brüsemeister, B. Bucciarelli, F. Budnik, P. Burgess, R. Burgon, A. Burlacu, D. Busonero, R. Buzzi, E. Caffau, J. Cambras, H. Campbell, R. Cancelliere, T. Cantat-Gaudin, T. Carlucci, J. M. Carrasco, M. Castellani, P. Charlot, J. Charnas, P. Charvet, F. Chassat, A. Chiavassa, M. Clotet, G. Cocozza, R. S. Collins, P. Collins, G. Costigan, F. Crifo, N. J. G. Cross, M. Crosta, C. Crowley, C. Dafonte, Y. Damerdji, A. Dapergolas, P. David, M. David, P. De Cat, F. de Felice, P. de Laverny, F. De Luise, R. De March, D. de Martino, R. de Souza, J. Debosscher, E. del Pozo, M. Delbo, A. Delgado, H. E. Delgado, F. di Marco, P. Di Matteo, S. Diakite, E. Distefano, C. Dolding, S. Dos Anjos, P. Drazinos, J. Durán, Y. Dzigan, E. Ecale, B. Edvardsson, H. Enke, M. Erdmann, D. Escolar, M. Espina, N. W. Evans, G. Eynard Bontemps, C. Fabre, M. Fabrizio, S. Faigler, A. J. Falcão, M. Farràs Casas, F. Faye, L. Federici, G. Fedorets, J. Fernández-Hernández, P. Fernique, A. Fienga, F. Figueras, F. Filippi, K. Findeisen, A. Fonti, M. Fouesneau, E. Fraile, M. Fraser, J. Fuchs, R. Furnell, M. Gai, S. Galleti, L. Galluccio, D. Garabato, F. García-Sedano, P. Garé, A. Garofalo, N. Garralda, P. Gavras, J. Gerssen, R. Geyer, G. Gilmore, S. Girona, G. Giuffrida, M. Gomes, A. González-Marcos, J. González-Núñez, J. J. González-Vidal, M. Granvik, A. Guerrier, P. Guillout, J. Guiraud, A. Gúrpide, R. Gutiérrez-Sánchez, L. P. Guy, R. Haigron, D. Hatzidimitriou, M. Haywood, U. Heiter, A. Helmi, D. Hobbs, W. Hofmann, B. Holl, G. Holland , J. A. S. Hunt, A. Hypki, V. Icardi, M. Irwin, G. Jevardat de Fombelle, P. Jofré, P. G. Jonker, A. Jorissen, F. Julbe, A. Karampelas, A. Kochoska, R. Kohley, K. Kolenberg, E. Kontizas, S. E. Koposov, G. Kordopatis, P. Koubsky, A. Kowalczyk, A. Krone-Martins, M. Kudryashova, I. Kull, R. K. Bachchan, F. Lacoste-Seris, A. F. Lanza, J. B. Lavigne, C. Le Poncin-Lafitte, Y. Lebreton, T. Lebzelter, S. Leccia, N. Leclerc, I. Lecoeur-Taibi, V. Lemaitre, H. Lenhardt, F. Leroux, S. Liao, E. Licata, H. E. P. Lindstrøm, T. A. Lister, E. Livanou, A. Lobel, W. Löffler, M. López, A. Lopez-Lozano, D. Lorenz, T. Loureiro, I. MacDonald, T. Magalhães Fernandes, S. Mana-

gau, R. G. Mann, G. Mantelet, O. Marchal, J. M. Marchant, M. Marconi, J. Marie, S. Marinoni, P. M. Marrese, G. Marschalkó, D. J. Marshall, J. M. Martín-Fleitas, M. Martino, N. Mary, G. Matijevič, T. Mazeh, P. J. McMillan, S. Messina, A. Mestre, D. Michalik, N. R. Millar, B. M. H. Miranda, D. Molina, R. Molinaro, M. Molinaro, L. Molnár, M. Moniez, P. Montegriffo, D. Monteiro, R. Mor, A. Mora, R. Morbidelli, T. Morel, S. Morgenthaler, T. Morley, D. Morris, A. F. Mulone, T. Muraveva, I. Musella, J. Narbonne, G. Nelemans, L. Nicastro, L. Noval, C. Ordénovic, J. Ordieres-Meré, P. Osborne, C. Pagani, I. Pagano, F. Pailler, H. Palacin, L. Palaversa, P. Parsons, T. Paulsen, M. Pecoraro, R. Pedrosa, H. Pentikäinen, J. Pereira, B. Pichon, A. M. Piersimoni, F. X. Pineau, E. Plachy, G. Plum, E. Poujoulet, A. Prša, L. Pulone, S. Ragaini, S. Rago, N. Rambaux, M. Ramos-Lerate, P. Ranalli, G. Rauw, A. Read, S. Regibo, F. Renk, C. Reylé, R. A. Ribeiro, L. Rimoldini, V. Ripepi, A. Riva, G. Rixon, M. Roelens, M. Romero-Gómez, N. Rowell, F. Royer, A. Rudolph, L. Ruiz-Dern, G. Sadowski, T. Sagristà Sellés, J. Sahlmann, J. Salgado, E. Salguero, M. Sarasso, H. Savietto, A. Schnorhk, M. Schultheis, E. Sciacca, M. Segol, J. C. Segovia, D. Segransan, E. Serpell, I. C. Shih, R. Smareglia, R. L. Smart, C. Smith, E. Solano, F. Solitro, R. Sordo, S. Soria Nieto, J. Souchay, A. Spagna, F. Spoto, U. Stampa, I. A. Steele, H. Steidelmüller, C. A. Stephenson, H. Stoev, F. F. Suess, M. Süveges, J. Surdej, L. Szabados, E. Szegedi-Elek, D. Tapiador, F. Taris, G. Tauran, M. B. Taylor, R. Teixeira, D. Terrett, B. Tingley, S. C. Trager, C. Turon, A. Ulla, E. Utrilla, G. Valentini, A. van Elteren, E. Van Hemelryck, M. van Leeuwen, M. Varadi, A. Vecchiato, J. Veljanoski, T. Via, D. Vicente, S. Vogt, H. Voss, V. Votruba, S. Voutsinas, G. Walmsley, M. Weiler, K. Weingrill, D. Werner, T. Wevers, G. Whitehead, Ł. Wyrzykowski, A. Yoldas, M. Žerjal, S. Zucker, C. Zurbach, T. Zwitter, A. Alecu, M. Allen, C. Allende Prieto, A. Amorim, G. Anglada-Escudé, V. Arsenijevic, S. Azaz, P. Balm, M. Beck, H. H. Bernstein, L. Bigot, A. Bijaoui, C. Blasco, M. Bonfigli, G. Bono, S. Boudreault, A. Bressan, S. Brown, P. M. Brunet, P. Bunclark, R. Buonanno, A. G. Butkevich, C. Carret, C. Carrion, L. Chemin, F. Chéreau, L. Corcione, E. Darmigny, K. S. de Boer, P. de Teodoro, P. T. de Zeeuw, C. Delle Luche, C. D. Domingues, P. Dubath, F. Fodor, B. Frézouls, A. Fries, D. Fustes, D. Fyfe, E. Gallardo, J. Gallegos, D. Gardiol, M. Gebran, A. Gomboc, A. Gómez, E. Grux, A. Gueguen, A. Heyrovsky, J. Hoar, G. Iannicola, Y. Isasi Parache, A. M. Janotto, E. Joliet, A. Jonckheere, R. Keil, D. W. Kim, P. Klagyivik, J. Klar, J. Knude, O. Kochukhov, I. Kolka, J. Kos, A. Kutka, V. Lainey, D. LeBouquin, C. Liu, D. Loreggia, V. V. Makarov, M. G. Marseille, C. Martayan, O. Martinez-Rubi, B. Massart, F. Meynadier, S. Mignot, U. Munari, A. T. Nguyen, T. Nordlander, P. Ocvirk, K. S. O'Flaherty, A. Olias Sanz, P. Ortiz, J. Osorio, D. Oszkiewicz, A. Ouzounis, M. Palmer, P. Park, E. Pasquato, C. Peltzer, J. Peralta, F. Péturaud, T. Pieniluoma, E. Pigozzi, J. Poels, G. Prat, T. Prod'homme, F. Raison, J. M. Rebordao, D. Risquez, B. Rocca-Volmerange, S. Rosen, M. I. Ruiz-Fuertes, F. Russo, S. Sembay, I. Serraller Vizcaino, A. Short, A. Siebert, H. Silva, D. Sinachopoulos, E. Slezak, M. Soffel, D. Sosnowska, V. Straižys, M. ter Linden, D. Terrell, S. Theil, C. Tiede, L. Troisi, P. Tsalmantza, D. Tur, M. Vaccari, F. Vachier, P. Valles, W. Van Hamme, L. Veltz, J. Virtanen, J. M. Wallut, R. Wichmann, M. I. Wilkinson, H. Ziaeepour, and S. Zschocke. The Gaia mission. A&A, 595:A1, Nov. 2016. doi: 10.1051/0004-6361/201629272. 1, C

A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017. 5.3

L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015. 4.2.5

L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural

networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 4.2.5

E. Gaztanaga, S. J. Schmidt, M. D. Schneider, and J. A. Tyson. Blending and obscuration in weak lensing magnification. *arXiv e-prints*, art. arXiv:2003.01047, Mar. 2020. 1, C

R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 6.2.1

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 6.2.1

X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In G. Gordon, D. Dunson, and M. Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL http://proceedings.mlr.press/v15/glorot11a.html. 4.2.3

R. E. González, R. P. Muñoz, and C. A. Hernández. Galaxy detection and identification using deep learning and data augmentation. *Astronomy and Computing*, 25:103–109, Oct. 2018. doi: 10.1016/j.ascom.2018.09.004. 6.3

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014a. 4.2.5

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org. 1, 4.2.5, 4.2.5, 4.2.5, 4.2.5, 4.3.3

I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013. 4.2.3

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014b. 4.2.5

M. J. Griffin, A. Abergel, A. Abreu, P. A. R. Ade, P. André, J.-L. Augueres, T. Babbedge, Y. Bae, T. Baillie, J.-P. Baluteau, M. J. Barlow, G. Bendo, D. Benielli, J. J. Bock, P. Bonhomme, D. Brisbin, C. Brockley-Blatt, M. Caldwell, C. Cara, N. Castro-Rodriguez, R. Cerulli, P. Chanial, S. Chen, E. Clark, D. L. Clements, L. Clerc, J. Coker, D. Communal, L. Conversi, P. Cox, D. Crumb, C. Cunningham, F. Daly, G. R. Davis, P. de Antoni, J. Delderfield, N. Devin, A. di Giorgio, I. Didschuns, K. Dohlen, M. Donati, A. Dowell, C. D. Dowell, L. Duband, L. Dumaye, R. J. Emery, M. Ferlet, D. Ferrand, J. Fontignie, M. Fox, A. Franceschini, M. Frerking, T. Fulton, J. Garcia, R. Gastaud, W. K. Gear, J. Glenn, A. Goizel, D. K. Griffin, T. Grundy, S. Guest, L. Guillemet, P. C. Hargrave, M. Harwit, P. Hastings, E. Hatziminaoglou, M. Herman, B. Hinde, V. Hristov, M. Huang, P. Imhof, K. J. Isaak, U. Israelsson, R. J. Ivison, D. Jennings, B. Kiernan, K. J. King, A. E. Lange, W. Latter, G. Laurent, P. Laurent, S. J. Leeks, E. Lellouch, L. Levenson, B. Li, J. Li, J. Lilienthal, T. Lim, S. J. Liu, N. Lu, S. Madden, G. Mainetti, P. Marliani, D. McKay, K. Mercier, S. Molinari, H. Morris, H. Moseley, J. Mulder, M. Mur, D. A. Naylor, H. Nguyen, B. O'Halloran, S. Oliver, G. Olofsson, H.-G. Olofsson, R. Orfei, M. J. Page, I. Pain, P. Panuzzo, A. Papageorgiou, G. Parks, P. Parr-Burman, A. Pearce, C. Pearson, I. Pérez-Fournon, F. Pinsard, G. Pisano, J. Podosek, M. Pohlen, E. T. Polehampton, D. Pouliquen, D. Rigopoulou, D. Rizzo, I. G.

Roseboom, H. Roussel, M. Rowan-Robinson, B. Rownd, P. Saraceno, M. Sauvage, R. Savage, G. Savini, E. Sawyer, C. Scharmberg, D. Schmitt, N. Schneider, B. Schulz, A. Schwartz, R. Shafer, D. L. Shupe, B. Sibthorpe, S. Sidher, A. Smith, A. J. Smith, D. Smith, L. Spencer, B. Stobie, R. Sudiwala, K. Sukhatme, C. Surace, J. A. Stevens, B. M. Swinyard, M. Trichas, T. Tourette, H. Triou, S. Tseng, C. Tucker, A. Turner, M. Vaccari, I. Valtchanov, L. Vigroux, E. Virique, G. Voellmer, H. Walker, R. Ward, T. Waskett, M. Weilert, R. Wesson, G. J. White, N. Whitehouse, C. D. Wilson, B. Winter, A. L. Woodcraft, G. S. Wright, C. K. Xu, A. Zavagno, M. Zemcov, L. Zhang, and E. Zonca. The Herschel-SPIRE instrument and its in-flight performance. A&A, 518:L3, July 2010. doi: 10.1051/0004-6361/201014519. 3.3, 5.3, 5.4.2

P. J. Grother. Nist special database 19 handprinted forms and characters database. *National Institute of Standards and Technology*, 1995. 4.3.2

D. Gruen, S. Seitz, and G. M. Bernstein. Implementation of Robust Image Artifact Removal in SWarp through Clipped Mean Stacking. PASP, 126(936):158, Feb. 2014. doi: 10.1086/675080. 5.2

D. Gruen, Y. Zhang, A. Palmese, B. Yanny, V. Busti, B. Hoyle, P. Melchior, C. J. Miller, E. Rozo, E. S. Rykoff, T. N. Varga, F. B. Abdalla, S. Allam, J. Annis, S. Avila, D. Brooks, D. L. Burke, A. Carnero Rosell, M. Carrasco Kind, J. Carretero, R. Cawthon, M. Crocce, C. E. Cunha, L. N. da Costa, C. Davis, J. De Vicente, S. Desai, H. T. Diehl, J. P. Dietrich, A. Drlica-Wagner, B. Flaugher, P. Fosalba, J. Frieman, J. García-Bellido, E. Gaztanaga, D. W. Gerdes, R. A. Gruendl, J. Gschwend, D. L. Hollowood, K. Honscheid, D. J. James, T. Jeltema, E. Krause, R. Kron, K. Kuehn, N. Kuropatkin, O. Lahav, M. Lima, H. Lin, M. A. G. Maia, J. L. Marshall, F. Menanteau, R. Miquel, R. L. C. Ogando, A. A. Plazas, A. K. Romer, V. Scarpine, I. Sevilla-Noarbe, M. Smith, M. Soares-Santos, F. Sobreira, E. Suchyta, M. E. C. Swanson, G. Tarle, D. Thomas, V. Vikram, A. R. Walker, and DES Collaboration. Dark Energy Survey Year 1 results: the effect of intracluster light on photometric redshifts for weak gravitational lensing. MNRAS, 488(3):4389–4399, Sept. 2019. doi: 10.1093/mnras/stz2036. 1, C

J. B. Hampshire II and B. Pearlmutter. Equivalence proofs for multi-layer perceptron classifiers and the bayesian discriminant function. In *Connectionist Models*, pages 159–172. Elsevier, 1991. 4.2.6, D

T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009. 4.11

R. Hausen and B. E. Robertson. Morpheus: A Deep Learning Framework for the Pixel-level Analysis of Astronomical Image Data. ApJS, 248(1):20, May 2020. doi: 10.3847/1538-4365/ab8868. 6.3

H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009. 4.2.6

K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 4.2.3

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4.3.3, A.1, A.2

K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1, 6.2.1, 6.3, C

D. O. Hebb. *The organization of behavior: a neuropsychological theory*. J. Wiley; Chapman & Hall, 1949. 4.2.1

D. Hendel and K. V. Johnston. Tidal debris morphology and the orbits of satellite galaxies. MNRAS, 454(3):2472–2485, Dec. 2015. doi: 10.1093/mnras/stv2035. 1, C

D. Herranz and J. L. Sanz. Matrix Filters for the Detection of Extragalactic Point Sources in Cosmic Microwave Background Images. *IEEE Journal of Selected Topics in Signal Processing*, 2(5):727–734, Nov. 2008. doi: 10.1109/JSTSP.2008.2005339. 3.1.2

D. Herranz, M. López-Caniego, J. L. Sanz, and J. González-Nuevo. A novel multifrequency technique for the detection of point sources in cosmic microwave background maps. MNRAS, 394(1):510–520, Mar. 2009. doi: 10.1111/j.1365-2966.2008.14336.x. 3.1.2

A. D. Herzog and G. Illingworth. The Structure of Globular Clusters. I. Direct Plane Automated Reduction Techniques. ApJS, 33:55, Jan. 1977. doi: 10.1086/190418. 3.1.1, 3.1.3, 3.1.5

C. Heymans, L. Van Waerbeke, L. Miller, T. Erben, H. Hildebrandt, H. Hoekstra, T. D. Kitching, Y. Mellier, P. Simon, C. Bonnett, J. Coupon, L. Fu, J. Harnois Déraps, M. J. Hudson, M. Kilbinger, K. Kuijken, B. Rowe, T. Schrabback, E. Semboloni, E. van Uitert, S. Vafaei, and M. Velander. CFHTLenS: the Canada-France-Hawaii Telescope Lensing Survey. MNRAS, 427(1):146–166, Nov. 2012. doi: 10.1111/j.1365-2966.2012.21952.x. 5, 5.2

G. E. Hinton. Learning translation invariant recognition in a massively parallel networks. In *International Conference on Parallel Architectures and Languages Europe*, pages 1–13. Springer, 1987. 4.2.5

G. E. Hinton. Deterministic boltzmann learning performs steepest descent in weight-space. *Neural computation*, 1(1):143–150, 1989. 4.2.4

G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012. 4.2.5, 4.3.3

M. P. Hobson and C. McLachlan. A Bayesian approach to discrete object detection in astronomical data sets. MNRAS, 338(3):765–784, Jan. 2003. doi: 10.1046/j.1365-8711.2003.06094.x. 3.3.4

A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. 4.2.5

J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982. 4.1

J. J. Hopfield. Learning algorithms and probability distributions in feed-forward and feed-back networks. *Proceedings of the national academy of sciences*, 84(23):8429–8433, 1987. 4.2.4

A. M. Hopkins, C. J. Miller, A. J. Connolly, C. Genovese, R. C. Nichol, and L. Wasserman. A New Source Detection Algorithm Using the False-Discovery Rate. AJ, 123(2):1086–1094, Feb. 2002. doi: 10.1086/338316. 3.1.1, 3.1.2, 3.1.4

K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4 (2):251–257, 1991. 4.2.2

K. Hornik, M. Stinchcombe, H. White, et al. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989. 4.2.2

P. V. Hough. Method and means for recognizing complex patterns, Dec. 18 1962. US Patent 3,069,654. 5.2

R. Hufnagel. Variations of atmospheric turbulence. *Topical Meeting on Optical Propagation through Turbulence, University of Colorado, Boulder, CO*, 1974. 2.4.3

N. Ienaka, K. Kawara, Y. Matsuoka, H. Sameshima, S. Oyabu, T. Tsujimoto, and B. A. Peterson. Diffuse Galactic Light in the Field of the Translucent High Galactic Latitude Cloud MBM32. ApJ, 767:80, Apr. 2013. doi: 10.1088/0004-637X/767/1/80. 5.4.2

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 4.2.7

M. Irwin. Automatic analysis of crowded fields. MNRAS, 214:575–604, June 1985. doi: 10.1093/mnras/214.4.575. 3.1.1, 3.1.2, 3.1.4, 3.1.5, 3.8

D. Ives. The INT Wide Field Camera. *IEEE Spectrum*, 16:20–21, Oct. 1998. 5.1

Ž. Ivezić, S. M. Kahn, J. A. Tyson, B. Abel, E. Acosta, R. Allsman, D. Alonso, Y. AlSayyad, S. F. Anderson, J. Andrew, J. R. P. Angel, G. Z. Angeli, R. Ansari, P. Antilogus, C. Araujo, R. Armstrong, K. T. Arndt, P. Astier, É. Aubourg, N. Auza, T. S. Axelrod, D. J. Bard, J. D. Barr, A. Barrau, J. G. Bartlett, A. E. Bauer, B. J. Bauman, S. Baumont, E. Bechtol, K. Bechtol, A. C. Becker, J. Becla, C. Beldica, S. Bellavia, F. B. Bianco, R. Biswas, G. Blanc, J. Blazek, R. D. Bland ford, J. S. Bloom, J. Bogart, T. W. Bond, M. T. Booth, A. W. Borgland, K. Borne, J. F. Bosch, D. Boutigny, C. A. Brackett, A. Bradshaw, W. N. Brand t, M. E. Brown, J. S. Bullock, P. Burchat, D. L. Burke, G. Cagnoli, D. Calabrese, S. Callahan, A. L. Callen, J. L. Carlin, E. L. Carlson, S. Chand rasekharan, G. Charles-Emerson, S. Chesley, E. C. Cheu, H.-F. Chiang, J. Chiang, C. Chirino, D. Chow, D. R. Ciardi, C. F. Claver, J. Cohen-Tanugi, J. J. Cockrum, R. Coles, A. J. Connolly, K. H. Cook, A. Cooray, K. R. Covey, C. Cribbs, W. Cui, R. Cutri, P. N. Daly, S. F. Daniel, F. Daruich, G. Daubard, G. Daues, W. Dawson, F. Delgado, A. Dellapenna, R. de Peyster, M. de Val-Borro, S. W. Digel, P. Doherty, R. Dubois, G. P. Dubois-Felsmann, J. Durech, F. Economou, T. Eifler, M. Eracleous, B. L. Emmons, A. Fausti Neto, H. Ferguson, E. Figueroa, M. Fisher-Levine, W. Focke, M. D. Foss, J. Frank, M. D. Freemon, E. Gangler, E. Gawiser, J. C. Geary, P. Gee, M. Geha, C. J. B. Gessner, R. R. Gibson, D. K. Gilmore, T. Glanzman, W. Glick, T. Goldina, D. A. Goldstein, I. Goodenow, M. L. Graham, W. J. Gressler, P. Gris, L. P. Guy, A. Guyonnet, G. Haller, R. Harris, P. A. Hascall, J. Haupt, F. Hernand ez, S. Herrmann, E. Hileman, J. Hoblitt, J. A. Hodgson, C. Hogan, J. D. Howard, D. Huang, M. E. Huffer, P. Ingraham, W. R. Innes, S. H. Jacoby, B. Jain, F. Jammes, M. J. Jee, T. Jenness, G. Jernigan, D. Jevremović, K. Johns, A. S. Johnson, M. W. G. Johnson, R. L. Jones, C. Juramy-Gilles, M. Jurić, J. S. Kalirai, N. J. Kallivayalil, B. Kalmbach, J. P. Kantor, P. Karst, M. M. Kasliwal, H. Kelly, R. Kessler, V. Kinnison, D. Kirkby, L. Knox, I. V. Kotov, V. L. Krabbendam, K. S. Krughoff, P. Kubánek, J. Kuczewski, S. Kulkarni, J. Ku, N. R. Kurita, C. S. Lage, R. Lambert, T. Lange, J. B. Langton, L. Le Guillou, D. Levine, M. Liang, K.-T. Lim, C. J. Lintott, K. E. Long, M. Lopez, P. J. Lotz, R. H. Lupton, N. B. Lust, L. A. MacArthur, A. Mahabal, R. Mand elbaum, T. W. Markiewicz, D. S. Marsh, P. J. Marshall, S. Marshall, M. May, R. McKercher, M. McQueen, J. Meyers, M. Migliore, M. Miller, D. J. Mills, C. Miraval, J. Moeyens, F. E. Moolekamp, D. G. Monet, M. Moniez, S. Monkewitz, C. Montgomery, C. B. Morrison, F. Mueller, G. P. Muller, F. Muñoz Arancibia, D. R. Neill, S. P. Newbry, J.-Y. Nief, A. Nomerotski, M. Nordby, P. O'Connor, J. Oliver, S. S. Olivier, K. Olsen, W. O'Mullane, S. Ortiz, S. Osier, R. E. Owen, R. Pain, P. E. Palecek, J. K. Parejko, J. B. Parsons, N. M. Pease, J. M. Peterson, J. R. Peterson, D. L. Petravick, M. E. Libby Petrick, C. E. Petry, F. Pierfederici, S. Pietrowicz,

R. Pike, P. A. Pinto, R. Plante, S. Plate, J. P. Plutchak, P. A. Price, M. Prouza, V. Radeka, J. Rajagopal, A. P. Rasmussen, N. Regnault, K. A. Reil, D. J. Reiss, M. A. Reuter, S. T. Ridgway, V. J. Riot, S. Ritz, S. Robinson, W. Roby, A. Roodman, W. Rosing, C. Roucelle, M. R. Rumore, S. Russo, A. Saha, B. Sassolas, T. L. Schalk, P. Schellart, R. H. Schindler, S. Schmidt, D. P. Schneider, M. D. Schneider, W. Schoening, G. Schumacher, M. E. Schwamb, J. Sebag, B. Selvy, G. H. Sembroski, L. G. Seppala, A. Serio, E. Serrano, R. A. Shaw, I. Shipsey, J. Sick, N. Silvestri, C. T. Slater, J. A. Smith, R. C. Smith, S. Sobhani, C. Soldahl, L. Storrie-Lombardi, E. Stover, M. A. Strauss, R. A. Street, C. W. Stubbs, I. S. Sullivan, D. Sweeney, J. D. Swinbank, A. Szalay, P. Takacs, S. A. Tether, J. J. Thaler, J. G. Thayer, S. Thomas, A. J. Thornton, V. Thukral, J. Tice, D. E. Trilling, M. Turri, R. Van Berg, D. Vanden Berk, K. Vetter, F. Virieux, T. Vucina, W. Wahl, L. Walkowicz, B. Walsh, C. W. Walter, D. L. Wang, S.-Y. Wang, M. Warner, O. Wiecha, B. Willman, S. E. Winters, D. Wittman, S. C. Wolff, W. M. Wood-Vasey, X. Wu, B. Xin, P. Yoachim, and H. Zhan. LSST: From Science Drivers to Reference Design and Anticipated Data Products. ApJ, 873(2):111, Mar. 2019. doi: 10.3847/1538-4357/ab042c. 1, 5.2, C

R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991. 4.2.3

N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002. 4.2.6

K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th international conference on computer vision*, pages 2146–2153. IEEE, 2009. 4.2.3

P. Jia, Q. Liu, and Y. Sun. Detection and Classification of Astronomical Targets with Deep Neural Networks in Wide-field Small Aperture Telescopes. AJ, 159(5):212, May 2020. doi: 10.3847/1538-3881/ab800a. 6.3

Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song. Neural style transfer: A review. *IEEE transactions on visualization and computer graphics*, 2019. 4.2.5

D. E. Jones, V. L. Kashyap, and D. A. van Dyk. Disentangling Overlapping Astronomical Sources Using Spatial and Spectral Information. ApJ, 808(2):137, Aug. 2015. doi: 10.1088/0004-637X/808/2/137. 3.3.4

R. Joseph, F. Courbin, and J. L. Starck. Multi-band morpho-Spectral Component Analysis Deblending Tool (MuSCADeT): Deblending colourful objects. A&A, 589:A2, May 2016. doi: 10.1051/0004-6361/201527923. 3.1.5

T. Kacprzak, J. Herbel, A. Nicola, R. Sgier, F. Tarsitano, C. Bruderer, A. Amara, A. Refregier, S. L. Bridle, A. Drlica-Wagner, D. Gruen, W. G. Hartley, B. Hoyle, L. F. Secco, J. Zuntz, J. Annis, S. Avila, E. Bertin, D. Brooks, E. Buckley-Geer, A. Carnero Rosell, M. Carrasco Kind, J. Carretero, L. N. da Costa, J. De Vicente, S. Desai, H. T. Diehl, P. Doel, J. García-Bellido, E. Gaztanaga, R. A. Gruendl, J. Gschwend, G. Gutierrez, D. L. Hollowood, K. Honscheid, D. J. James, M. Jarvis, M. Lima, M. A. G. Maia, J. L. Marshall, P. Melchior, F. Menanteau, R. Miquel, F. Paz-Chinchón, A. A. Plazas, E. Sanchez, V. Scarpine, S. Serrano, I. Sevilla-Noarbe, M. Smith, E. Suchyta, M. E. C. Swanson, G. Tarle, V. Vikram, J. Weller, and DES Collaboration. Monte Carlo control loops for cosmic shear cosmology with DES Year 1 data. Phys. Rev. D, 101(8):082003, Apr. 2020. doi: 10.1103/PhysRevD.101.082003. 1, C

N. Kaiser, G. Squires, and T. Broadhurst. A Method for Weak Lensing Observations. ApJ, 449:460, Aug. 1995. doi: 10.1086/176071. 3.3

G. Kauffmann, S. D. M. White, and B. Guiderdoni. The formation and evolution of galaxies within merging dark matter haloes. MNRAS, 264:201–218, Sept. 1993. doi: 10.1093/mnras/264.1.201. 1, C

S. Kawanomoto, Y. Komiyama, and M. Yagi. Ghost busters: Subaru/hsc ghost analysis (2) & removal arc ghosts. In *Subaru Users' Meeting FY2016*, 2016a. 5.2

Y. Kawanomoto, M. Yagi, and S. Kawanomoto. Ghost busters: Subaru/hsc ghost analysis (1) cometary ghosts. In *Subaru Users' Meeting FY2016*, 2016b. 5.2

M. Kendall and A. Stuart. *The advanced theory of statistics. Vol.1: Distribution theory.* 1977. 3.1.1

S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems*, 29(8):3573–3587, 2017. 4.2.6

A. Khotanzad and C. Chung. Application of multi-layer perceptron neural networks to vision problems. *Neural Computing & Applications*, 7(3):249–259, 1998. 4.2.7

A. Khotanzad and J.-H. Liou. Recognition and pose estimation of unoccluded three-dimensional objects from a two-dimensional perspective view by banks of neural networks. *IEEE Transactions on Neural networks*, 7(4):897–906, 1996. 4.2.7

I. R. King. The Profile of a Star Image. PASP, 83(492):199, Apr. 1971. doi: 10.1086/129100. 2.4.4, 3.1.5

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5.4.2, 5.5.3, 5.5.4, 6.5.6

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 4.2, 6.3

J. Knollmüller, P. Frank, and T. A. Enßlin. Separating diffuse from point-like sources - a Bayesian approach. *arXiv e-prints*, art. arXiv:1804.05591, Apr. 2018. 1, C

T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69, 1982. 4.1, 5.3.2

A. Kolmogorov. The Local Structure of Turbulence in Incompressible Viscous Fluid for Very Large Reynolds' Numbers. *Akademiia Nauk SSSR Doklady*, 30:301–305, Jan. 1941a. 2.4.3

A. N. Kolmogorov. Dissipation of Energy in Locally Isotropic Turbulence. *Akademiia Nauk SSSR Doklady*, 32:16, Apr. 1941b. 2.4.3

B. Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016. 4.2.6

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1, 4.2.5, 4.3.3, 4.16, 7, C, D, E

R. G. Kron. Photometry of a complete sample of faint galaxies. ApJS, 43:305–325, June 1980. doi: 10.1086/190669. 3.1.1, 3.1.3

H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 6.2.3

K. Kuijken, R. Bender, E. Cappellaro, B. Muschielok, A. Baruffolo, E. Cascone, O. Iwert, W. Mitsch, H. Nicklas, E. A. Valentijn, D. Baade, K. G. Begeman, A. Bortolussi, D. Boxhoorn, F. Christen, E. R. Deul, C. Geimer, L. Greggio, R. Harke, R. Häfner, G. Hess, H.-J. Hess, U. Hopp, I. Ilijevski, G. Klink, H. Kravcar, J. L. Lizon, C. E. Magagna, P. Müller, R. Niemeczek, L. de Pizzol, H. Poschmann, K. Reif, R. Rengelink, J. Reyes, A. Silber, and W. Wellem. OmegaCAM: the 16k×16k CCD camera for the VLT survey telescope. *The Messenger*, 110:15–18, Dec. 2002. 5.1

V. Kulikov, V. Yurchenko, and V. Lempitsky. Instance Segmentation by Deep Coloring. *arXiv e-prints*, art. arXiv:1807.10007, July 2018. 6.2.3

B. Kwolek. Face detection using convolutional neural networks and gabor filters. In *International Conference on Artificial Neural Networks*, pages 551–556. Springer, 2005. 4.3.2

D. Lang, D. W. Hogg, K. Mierle, M. Blanton, and S. Roweis. Astrometry.net: Blind Astrometric Calibration of Arbitrary Astronomical Images. AJ, 139(5):1782–1800, May 2010. doi: 10.1088/0004-6256/139/5/1782. 3.1.1, 3.1.3

F. Lanusse, P. Melchior, and F. Moolekamp. Hybrid Physical-Deep Learning Model for Astronomical Inverse Problems. *arXiv e-prints*, art. arXiv:1912.03980, Dec. 2019. 6.3

F. Lanusse, R. Mandelbaum, S. Ravanbakhsh, C.-L. Li, P. Freeman, and B. Poczos. Deep Generative Models for Galaxy Image Simulations. *arXiv e-prints*, art. arXiv:2008.03833, Aug. 2020. 6.5.1

B. M. Lasker, C. R. Sturch, B. J. McLean, J. L. Russell, H. Jenkner, and M. M. Shara. The Guide Star Catalog. I. Astronomical Foundations and Image Processing. AJ, 99:2019, June 1990. doi: 10.1086/115483. 3.1.1, 3.1.4, 3.1.5

R. Laureijs, P. Gondoin, L. Duvet, G. Saavedra Criado, J. Hoar, J. Amiaux, J. L. Auguères, R. Cole, M. Cropper, A. Ealet, P. Ferruit, I. Escudero Sanz, K. Jahnke, R. Kohley, T. Maciaszek, Y. Mellier, T. Oosterbroek, F. Pasian, M. Sauvage, R. Scaramella, M. Sirianni, and L. Valenziano. Euclid: ESA's mission to map the geometry of the dark universe. In *Space Telescopes and Instrumentation 2012: Optical, Infrared, and Millimeter Wave*, volume 8442 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, page 84420T, Sept. 2012. doi: 10.1117/12.926496. 1, C

S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113, 1997. 4.3.2

D. Lazzati, S. Campana, P. Rosati, M. R. Panzera, and G. Tagliaferri. The Brera Multiscale Wavelet ROSAT HRI Source Catalog. I. The Algorithm. ApJ, 524(1):414–422, Oct. 1999. doi: 10.1086/307788. 3.1.1, 3.3.2

Y. Le Cun. Learning process in an asymmetric threshold network. In *Disordered systems and biological organization*, pages 233–240. Springer, 1986. 4.2.2

O. Le Fevre, A. Bijaoui, G. Mathez, J. P. Picat, and G. Lelievre. Electronographic BV photometry of three distant clusters of galaxies. A&A, 154:92–99, Jan. 1986. 3.1.1, 3.1.4, 3.1.5

Y. LeCun. A learning scheme for asymmetric threshold networks. *Proceedings of COGNITIVA*, 85(537):599–604, 1985a. 4.2.2

Y. LeCun. Une procedure d'apprentissage ponr reseau a seuil asymetrique. *Proceedings of Cognitiva 85*, pages 599–604, 1985b. 4.2.2

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 4.3.2

Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990. 4.1, 4.3.2

Y. LeCun, Y. Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995. 1, 4.3.2, 7, C, D, E

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 4.14, 4.1

Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012. 4.2.7

H. Lee, P. Pham, Y. Largman, and A. Y. Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in neural information processing systems*, pages 1096–1104, 2009. 4.3.3

J. Lemley, S. Bazrafkan, and P. Corcoran. Smart augmentation learning an optimal data augmentation strategy. *Ieee Access*, 5:5858–5869, 2017. 4.2.5

T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*, 2017. 5.5.2

Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang. Large-scale image classification: Fast feature extraction and svm training. In *CVPR 2011*, pages 1689–1696. IEEE, 2011. 4.3.3

W. A. Little. The existence of persistent states in the brain. *Mathematical biosciences*, 19(1-2): 101–120, 1974. 4.1

S. Lombardo, F. Prada, E. Hugot, S. Basa, J. M. Bautista, S. Boissier, A. Boselli, A. Bosma, J. C. Cuillandre, P. A. Duc, M. Ferrari, N. Grosso, L. Izzo, K. Joaquina, Junais, J. Koda, A. Lamberts, G. R. Lemaitre, A. Longobardi, D. Martínez-Delgado, E. Muslimov, J. L. Ortiz, E. Perez, D. Porquet, B. Sicardy, and P. Vola. CASTLE: performances and science cases. *arXiv e-prints*, art. arXiv:2006.13956, June 2020. 1, C

J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 5.3.1

K. S. Long, S. M. Baggett, and J. W. MacKenty. Persistence in the WFC3 IR Detector: an Improved Model Incorporating the Effects of Exposure Time. Technical report, Sept. 2015. 5.3, 5.4.2, 5.4.2

V. Lukic, F. De Gasperin, and M. Brüggen. Autosource: Radio-astronomical source-finding with convolutional autoencoders. *arXiv preprint arXiv:1910.03631*, 2019. 6.3

A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013. 4.2.3

D. J. MacKay. Bayesian modeling and neural networks. *PhD thesis, Dept. of Computation and Neural Systems, CalTech*, 1992. 4.2.5

S. J. Maddox and L. Dunne. MADX - a simple technique for source detection and measurement using multiband imaging from the Herschel-ATLAS survey. MNRAS, 493(2):2363–2372, Apr. 2020. doi: 10.1093/mnras/staa458. 3.1.2

E. A. Magnier and J.-C. Cuillandre. The Elixir System: Data Characterization and Calibration at the Canada-France-Hawaii Telescope. PASP, 116:449–464, May 2004. doi: 10.1086/420756. 5.4.1

D. Makovoz. Nonlinear matched filtering for point source detection. In E. R. Dougherty, J. T. Astola, and K. O. Egiazarian, editors, Proc. SPIE, volume 5672 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 358–369, Mar. 2005. doi: 10.1117/12.587283. 3.1.2

D. Makovoz and F. R. Marleau. Point-Source Extraction with MOPEX. PASP, 117(836):1113–1128, Oct. 2005. doi: 10.1086/432977. 3.1.1, 3.1.2, 3.1.5, 3.1.6

R. Mandelbaum, F. Lanusse, A. Leauthaud, R. Armstrong, M. Simet, H. Miyatake, J. E. Meyers, J. Bosch, R. Murata, S. Miyazaki, and M. Tanaka. Weak lensing shear calibration with simulations of the HSC survey. MNRAS, 481(3):3170–3195, Dec. 2018. doi: 10.1093/mnras/sty2420. 6.5.1

M. Masias, J. Freixenet, X. Lladó, and M. Peracaula. A review of source detection approaches in astronomical images. MNRAS, 422(2):1674–1689, May 2012. doi: 10.1111/j.1365-2966.2012.20742.x. 1

G. Matheron and J. Serra. The birth of mathematical morphology. In *Proc. 6th Intl. Symp. Mathematical Morphology*, pages 1–16. Sydney, Australia, 2002. 3.2

B. W. Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, 1975. 5.4.2, 5.6.1

W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943. 4.2.1, D

C. McCully and M. Tewes. Astro-SCRAPPY: Speedy Cosmic Ray Annihilation Package in Python, July 2019. 5.2, 5.4.1, 5.6.1, 5.6.1, D

P. Melchior, E. Sheldon, A. Drlica-Wagner, E. S. Rykoff, T. M. C. Abbott, F. B. Abdalla, S. Allam, A. Benoit-Lévy, D. Brooks, E. Buckley-Geer, A. Carnero Rosell, M. Carrasco Kind, J. Carretero, M. Crocce, C. B. D'Andrea, L. N. da Costa, S. Desai, P. Doel, A. E. Evrard, D. A. Finley, B. Flaugher, J. Frieman, E. Gaztanaga, D. W. Gerdes, D. Gruen, R. A. Gruendl, K. Honscheid, D. J. James, M. Jarvis, K. Kuehn, T. S. Li, M. A. G. Maia, M. March, J. L. Marshall, B. Nord, R. Ogando, A. A. Plazas, A. K. Romer, E. Sanchez, V. Scarpine, I. Sevilla-Noarbe, R. C. Smith, M. Soares-Santos, E. Suchyta, M. E. C. Swanson, G. Tarle, V. Vikram, A. R. Walker, W. Wester, and Y. Zhang. Crowdsourcing quality control for Dark Energy Survey images. *Astronomy and Computing*, 16:99–108, July 2016. doi: 10.1016/j.ascom.2016.04.003. 5

P. Melchior, F. Moolekamp, M. Jerdee, R. Armstrong, A. L. Sun, J. Bosch, and R. Lupton. SCARLET: Source separation in multi-band images by Constrained Matrix Factorization. *Astronomy and Computing*, 24:129, July 2018. doi: 10.1016/j.ascom.2018.07.001. 3.1.5

J. B. Melin, J. G. Bartlett, and J. Delabrouille. Catalog extraction in SZ cluster surveys: a matched filter approach. A&A, 459(2):341–352, Nov. 2006. doi: 10.1051/0004-6361:20065034. 3.1.2

M. R. Metzger, G. A. Luppino, and S. Miyazaki. The UH 8K CCD Mosaic Camera. In *American Astronomical Society Meeting Abstracts*, volume 187 of *American Astronomical Society Meeting Abstracts*, page 73.05, Dec. 1995. 5.1

K. J. Mighell. The Personal Astronomical Workstation Photometric Reduction Package. In *ESO/ST-ECF Data Analysis Workshop*, page 197, Jan. 1989a. 3.1.2, 3.1.3, 3.1.5

K. J. Mighell. Accurate stellar photometry in crowded fields. MNRAS, 238:807–833, May 1989b. doi: 10.1093/mnras/238.3.807. 3.1.3, 3.1.5

K. J. Mighell. Algorithms for CCD Stellar Photometry. In D. M. Mehringer, R. L. Plante, and D. A. Roberts, editors, *Astronomical Data Analysis Software and Systems VIII*, volume 172 of *Astronomical Society of the Pacific Conference Series*, page 317, Jan. 1999. 3.1.2, 3.1.3

A. Mikołajczyk and M. Grochowski. Data augmentation for improving deep learning in image classification problem. In *2018 international interdisciplinary PhD workshop (IIPhDW)*, pages 117–122. IEEE, 2018. 4.2.5

J. W. Miller, R. Goodman, and P. Smyth. Objective functions for probability estimation. 1991. 4.2.6

M. Minsky and S. Papert. Perceptron: an introduction to computational geometry. *The MIT Press, Cambridge, expanded edition*, 19(88):2, 1969. 4.2.1, 4.2.1, 4.2.1

M. A. Miville-Deschênes, P. A. Duc, F. Marleau, J. C. Cuillandre, P. Didelon, S. Gwyn, and E. Karabal. Probing interstellar turbulence in cirrus with deep optical imaging: no sign of energy dissipation at 0.01 pc scale. A&A, 593:A4, Aug. 2016. doi: 10.1051/0004-6361/201628503. 1, 5.4.2, C

S. Miyazaki, Y. Komiyama, S. Kawanomoto, Y. Doi, H. Furusawa, T. Hamana, Y. Hayashi, H. Ikeda, Y. Kamata, H. Karoji, M. Koike, T. Kurakami, S. Miyama, T. Morokuma, F. Nakata, K. Namikawa, H. Nakaya, K. Nariai, Y. Obuchi, Y. Oishi, N. Okada, Y. Okura, P. Tait, T. Takata, Y. Tanaka, M. Tanaka, T. Terai, D. Tomono, F. Uraguchi, T. Usuda, Y. Utsumi, Y. Yamada, H. Yamanoi, H. Aihara, H. Fujimori, S. Mineo, H. Miyatake, M. Oguri, T. Uchida, M. M. Tanaka, N. Yasuda, M. Takada, H. Murayama, A. J. Nishizawa, N. Sugiyama, M. Chiba, T. Futamase, S.-Y. Wang, H.-Y. Chen, P. T. P. Ho, E. J. Y. Liaw, C.-F. Chiu, C.-L. Ho, T.-C. Lai, Y.-C. Lee, D.-Z. Jeng, S. Iwamura, R. Armstrong, S. Bickerton, J. Bosch, J. E. Gunn, R. H. Lupton, C. Loomis, P. Price, S. Smith, M. A. Strauss, E. L. Turner, H. Suzuki, Y. Miyazaki, M. Muramatsu, K. Yamamoto, M. Endo, Y. Ezaki, N. Ito, N. Kawaguchi, S. Sofuku, T. Taniike, K. Akutsu, N. Dojo, K. Kasumi, T. Matsuda, K. Imoto, Y. Miwa, M. Suzuki, K. Takeshi, and H. Yokota. Hyper Suprime-Cam: System design and verification of image quality. PASJ, 70: S1, Jan. 2018. doi: 10.1093/pasj/psx063. 5.1

A. F. J. Moffat. A Theoretical Investigation of Focal Stellar Images in the Photographic Emulsion and Application to Photographic Photometry. A&A, 3:455, Dec. 1969. 3.1.5

B. Moore, S. Ghigna, F. Governato, G. Lake, T. Quinn, J. Stadel, and P. Tozzi. Dark Matter Substructure within Galactic Halos. ApJ, 524(1):L19–L22, Oct. 1999. doi: 10.1086/312287. 1, C

N. Morgan and H. Bourlard. Generalization and parameter estimation in feedforward nets: Some experiments. In *Advances in neural information processing systems*, pages 630–637, 1990. 4.2.5

E. Morganson, R. A. Gruendl, F. Menanteau, M. Carrasco Kind, Y.-C. Chen, G. Daues, A. Drlica-Wagner, D. N. Friedel, M. Gower, M. W. G. Johnson, M. D. Johnson, R. Kessler, F. Paz-Chinchón, D. Petravick, C. Pond, B. Yanny, S. Allam, R. Armstrong, W. Barkhouse, K. Bechtol, A. Benoit-Lévy, G. M. Bernstein, E. Bertin, E. Buckley-Geer, R. Covarrubias, S. Desai, H. T. Diehl, D. A. Goldstein, D. Gruen, T. S. Li, H. Lin, J. Marriner, J. J. Mohr, E. Neilsen, C.-C. Ngeow, K. Paech, E. S. Rykoff, M. Sako, I. Sevilla-Noarbe, E. Sheldon, F. Sobreira, D. L. Tucker, W. Wester, and DES Collaboration. The Dark Energy Survey Image Processing Pipeline. PASP, 130(7):074501, July 2018. doi: 10.1088/1538-3873/aab4ef. 5.2, D

R. Müller, S. Kornblith, and G. E. Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems*, pages 4696–4705, 2019. 4.2.5

V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077. 4.2.3

R. Nanni, R. Gilli, C. Vignali, M. Mignoli, A. Peca, S. Marchesi, M. Annunziatella, M. Brusa, F. Calura, N. Cappelluti, M. Chiaberge, A. Comastri, K. Iwasawa, G. Lanzuisi, E. Liuzzo, D. Marchesini, I. Prandoni, P. Tozzi, F. Vito, G. Zamorani, and C. Norman. The deep Chandra survey in the SDSS J1030+0524 field. A&A, 637:A52, May 2020. doi: 10.1051/0004-6361/202037914. 3.3.2

B. Newell and J. O'Neil, Earl J. The Reduction of Panoramic Photometry 1. Two Search Algorithms. PASP, 89:925, Dec. 1977. doi: 10.1086/130248. 3.1.3, 3.1.5

G. Nir, B. Zackay, and E. O. Ofek. Optimal and efficient streak detection in astronomical images. *arXiv preprint arXiv:1806.04204*, 2018. 5.2

M. Nonino, E. Bertin, L. da Costa, E. Deul, T. Erben, L. Olsen, I. Prandoni, M. Scodeggio, A. Wicenec, R. Wichmann, C. Benoist, W. Freudling, M. D. Guarnieri, I. Hook, R. Hook, R. Mendez, S. Savaglio, D. Silva, and R. Slijkhuis. ESO Imaging Survey. I. Description of the survey, data reduction and reliability of the data. A&AS, 137:51–74, May 1999. doi: 10.1051/aas:1999473. 4.3.2

R. Novak and Y. Nikulin. Improving the neural algorithm of artistic style. *arXiv preprint arXiv:1605.04603*, 2016. 4.2.5

S. J. Nowlan and G. E. Hinton. Simplifying neural networks by soft weight-sharing. *Neural computation*, 4(4):473–493, 1992. 4.2.5

B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996. 3.3.3

A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016. 6.3

C. Ordénovic, C. Surace, B. Torrésani, and A. Llébaria. Detection of glitches and signal reconstruction using Hölder and wavelet analysis. *Statistical Methodology*, 5:373–386, July 2008. doi: 10.1016/j.stamet.2008.01.005. 5.2

M. Osadchy, Y. L. Cun, and M. L. Miller. Synergistic face detection and pose estimation with energy-based models. *Journal of Machine Learning Research*, 8(May):1197–1215, 2007. 4.3.2

M. Paillassa and E. Bertin. Deblending in Crowded Star Fields Using Convolutional Neural Networks. In M. Molinaro, K. Shortridge, and F. Pasian, editors, *Astronomical Data Analysis Software and Systems XXVI*, volume 521 of *Astronomical Society of the Pacific Conference Series*, page 382, Oct. 2019. 6.1, 6.3

M. Paillassa, E. Bertin, and H. Bouy. MAXIMASK and MAXITRACK: Two new tools for identifying contaminants in astronomical images using convolutional neural networks. A&A, 634:A48, Feb. 2020. doi: 10.1051/0004-6361/201936345. 5, 5.1, 5.2, 5.20, 5.3, 5.4.2, 5.4.2, 5.4.2, 5.4.2, 5.31, 5.32, 5.36, 5.41, 5.46, 7, B.1, B.2, B.3, D, E

D. Paranjpye, A. Mahabal, A. N. Ramaprakash, G. V. Panopoulou, K. Cleary, A. C. S. Readhead, D. Blinov, and K. Tassis. Eliminating artefacts in polarimetric images using deep learning. MNRAS, 491(4):5151–5157, Feb. 2020. doi: 10.1093/mnras/stz3250. 5.3.2

K. Pearson. X. contributions to the mathematical theory of evolution.—ii. skew variation in homogeneous material. *Philosophical Transactions of the Royal Society of London.(A.)*, (186): 343–414, 1895. 3.1.1

K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. 4.2.7

A. J. Penny. Electronographic UBV photometry of close visual double stars. MNRAS, 187: 829–837, June 1979. doi: 10.1093/mnras/187.4.829. 3.1.5

A. J. Penny and R. J. Dickens. CCD photometry of the globular cluster NGC 6752. MNRAS, 220:845–867, June 1986. doi: 10.1093/mnras/220.4.845. 3.1.5

M. Peracaula, X. Lladó, J. Freixenet, A. Oliver, A. Torrent, J. M. Paredes, and J. Martí. *Segmentation and Detection of Extended Structures in Low Frequency Astronomical Surveys using Hybrid Wavelet Decomposition*, volume 442 of *Astronomical Society of the Pacific Conference Series*, page 151. 2011. 3.3.2

B. Perret, S. Lefèvre, and C. Collet. A robust hit-or-miss transform for template matching applied to very noisy astronomical images. *Pattern Recognition*, 42(11):2470–2480, 2009. 3.1.1, 3.2.2

B. Perret, S. Lefevre, C. Collet, and É. Slezak. Connected component trees for multivariate image processing and applications in astronomy. In *2010 20th International Conference on Pattern Recognition*, pages 4089–4092. IEEE, 2010. 3.2.2

R. A. Peters. A new algorithm for image noise reduction using mathematical morphology. *IEEE transactions on Image Processing*, 4(5):554–568, 1995. 3.2.2

J. R. Peterson, J. G. Jernigan, S. M. Kahn, A. P. Rasmussen, E. Peng, Z. Ahmad, J. Bankert, C. Chang, C. Claver, D. K. Gilmore, E. Grace, M. Hannel, M. Hodge, S. Lorenz, A. Lupu, A. Meert, S. Nagarajan, N. Todd, A. Winans, and M. Young. Simulation of Astronomical Images from Optical Survey Telescopes Using a Comprehensive Photon Monte Carlo Approach. ApJS, 218(1):14, May 2015. doi: 10.1088/0067-0049/218/1/14. 2.4

G. L. Pilbratt, J. R. Riedinger, T. Passvogel, G. Crone, D. Doyle, U. Gageur, A. M. Heras, C. Jewell, L. Metcalfe, S. Ott, and M. Schmidt. Herschel Space Observatory. An ESA facility for

far-infrared and submillimetre astronomy. A&A, 518:L1, July 2010. doi: 10.1051/0004-6361/201014759. 5.3, 5.4.2

D. C. Plaut et al. Experiments on learning by back propagation. 1986. 4.2.5

B. Poole, J. Sohl-Dickstein, and S. Ganguli. Analyzing noise in autoencoders and deep networks. *arXiv preprint arXiv:1406.1831*, 2014. 4.2.5

A. Popowicz and B. Smolka. A method of complex background estimation in astronomical images. MNRAS, 452(1):809–823, Sept. 2015. doi: 10.1093/mnras/stv1320. 1, C

A. Popowicz, A. R. Kurek, and Z. Filus. Bad Pixel Modified Interpolation for Astronomical Images. PASP, 125(931):1119, Sept. 2013. doi: 10.1086/673179. 1, C

S. K. N. Portillo, B. C. G. Lee, T. Daylan, and D. P. Finkbeiner. Improved Point-source Detection in Crowded Fields Using Probabilistic Cataloging. AJ, 154(4):132, Oct. 2017. doi: 10.3847/1538-3881/aa8565. 3.3.4

L. Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 1998. 4.2.5

A. M. Price-Whelan, B. M. Sipőcz, H. M. Günther, P. L. Lim, S. M. Crawford, S. Conseil, D. L. Shupe, M. W. Craig, N. Dencheva, A. Ginsburg, J. T. VanderPlas, L. D. Bradley, D. Pérez-Suárez, M. de Val-Borro, P. Paper Contributors, T. L. Aldcroft, K. L. Cruz, T. P. Robitaille, E. J. Tollerud, A. Coordination Committee, C. Ardelean, T. Babej, Y. P. Bach, M. Bachetti, A. V. Bakanov, S. P. Bamford, G. Barentsen, P. Barmby, A. Baumbach, K. L. Berry, F. Biscani, M. Boquien, K. A. Bostroem, L. G. Bouma, G. B. Brammer, E. M. Bray, H. Breytenbach, H. Buddelmeijer, D. J. Burke, G. Calderone, J. L. Cano Rodríguez, M. Cara, J. V. M. Cardoso, S. Cheedella, Y. Copin, L. Corrales, D. Crichton, D. D'Avella, C. Deil, É. Depagne, J. P. Dietrich, A. Donath, M. Droettboom, N. Earl, T. Erben, S. Fabbro, L. A. Ferreira, T. Finethy, R. T. Fox, L. H. Garrison, S. L. J. Gibbons, D. A. Goldstein, R. Gommers, J. P. Greco, P. Greenfield, A. M. Groener, F. Grollier, A. Hagen, P. Hirst, D. Homeier, A. J. Horton, G. Hosseinzadeh, L. Hu, J. S. Hunkeler, Ž. Ivezić, A. Jain, T. Jenness, G. Kanarek, S. Kendrew, N. S. Kern, W. E. Kerzendorf, A. Khvalko, J. King, D. Kirkby, A. M. Kulkarni, A. Kumar, A. Lee, D. Lenz, S. P. Littlefair, Z. Ma, D. M. Macleod, M. Mastropietro, C. McCully, S. Montagnac, B. M. Morris, M. Mueller, S. J. Mumford, D. Muna, N. A. Murphy, S. Nelson, G. H. Nguyen, J. P. Ninan, M. Nöthe, S. Ogaz, S. Oh, J. K. Parejko, N. Parley, S. Pascual, R. Patil, A. A. Patil, A. L. Plunkett, J. X. Prochaska, T. Rastogi, V. Reddy Janga, J. Sabater, P. Sakurikar, M. Seifert, L. E. Sherbert, H. Sherwood-Taylor, A. Y. Shih, J. Sick, M. T. Silbiger, S. Singanamalla, L. P. Singer, P. H. Sladen, K. A. Sooley, S. Sornarajah, O. Streicher, P. Teuben, S. W. Thomas, G. R. Tremblay, J. E. H. Turner, V. Terrón, M. H. van Kerkwijk, A. de la Vega, L. L. Watkins, B. A. Weaver, J. B. Whitmore, J. Woillez, V. Zabalza, and A. Contributors. The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package. AJ, 156:123, Sept. 2018. doi: 10.3847/1538-3881/aabc4f. 5.7

W. Pych. A Fast Algorithm for Cosmic-Ray Removal from Single Images. PASP, 116(816):148–153, Feb. 2004. doi: 10.1086/381786. 5.2

G. Qiu, M. Varley, and T. Terrell. Image compression by edge pattern learning using multilayer perceptrons. *Electronics letters*, 29(7):601–603, 1993. 4.2.7

G. D. Racca, R. Laureijs, L. Stagnaro, J.-C. Salvignol, J. Lorenzo Alvarez, G. Saavedra Criado, L. Gaspar Venancio, A. Short, P. Strada, T. Bönke, C. Colombo, A. Calvi, E. Maiorano,

O. Piersanti, S. Prezelus, P. Rosato, J. Pinel, H. Rozemeijer, V. Lesna, P. Musi, M. Sias, A. Anselmi, V. Cazaubiel, L. Vaillon, Y. Mellier, J. Amiaux, M. Berthé, M. Sauvage, R. Azzollini, M. Cropper, S. Pottinger, K. Jahnke, A. Ealet, T. Maciaszek, F. Pasian, A. Zacchei, R. Scaramella, J. Hoar, R. Kohley, R. Vavrek, A. Rudolph, and M. Schmidt. The Euclid mission design. In *Space Telescopes and Instrumentation 2016: Optical, Infrared, and Millimeter Wave*, volume 9904 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, page 99040O, July 2016. doi: 10.1117/12.2230762. 1, 5.2, C

R. Racine. The Telescope Point Spread Function. PASP, 108:699, Aug. 1996. doi: 10.1086/133788. 2.4.3, 2.4.4

J. Radon. Uber die bestimmung von funktionen durch ihre integralwerte langs gewissez mannigfaltigheiten, ber. *Verh. Sachs. Akad. Wiss. Leipzig, Math Phys Klass*, 69, 1917. 5.2

J. Radon. On the determination of functions from their integral values along certain manifolds. *IEEE transactions on medical imaging*, 5(4):170–176, 1986. 5.2

J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 6.2.2

J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 6.2.2

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1, 6.2.2, 6.2, 6.3, C

S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv e-prints*, art. arXiv:1506.01497, June 2015. 1, 6.2.1, 6.1, 6.3, C

J. P. Rheault, N. P. Mondrik, D. L. DePoy, J. L. Marshall, and N. B. Suntzeff. Spectrophotometric calibration of the Swope and duPont telescopes for the Carnegie supernova project 2. In *Ground-based and Airborne Instrumentation for Astronomy V*, volume 9147 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, page 91475L, Aug. 2014. doi: 10.1117/12.2063560. 5.1

J. E. Rhoads. Cosmic-Ray Rejection by Linear Filtering of Single Images. PASP, 112(771):703–710, May 2000. doi: 10.1086/316559. 5.2

M. D. Richard and R. P. Lippmann. Neural network classifiers estimate bayesian a posteriori probabilities. *Neural computation*, 3(4):461–483, 1991. 4.2.6, D

H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951. 4.2.1

F. Roddier. The effects of atmospheric turbulence in optical astronomy. *Progess in Optics*, 19:281–376, Jan. 1981. doi: 10.1016/S0079-6638(08)70204-X. 2.4, 2.4.3, 2.4.3

R. Rojas. A short proof of the posterior probability property of classifier neural networks. *Neural Computation*, 8(1):41–43, 1996. doi: 10.1162/neco.1996.8.1.41. URL https://doi.org/10.1162/neco.1996.8.1.41. 4.2.6, D

B. Romera-Paredes and P. H. S. Torr. Recurrent instance segmentation. In *European conference on computer vision*, pages 312–329. Springer, 2016. 6.2.3

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1, 5.3.1, 5.4.2, 5.5.1, 6.4.4, C, D

F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958. 4.2.1, 4.2.1, 4.2.1, 4.2.1, 4.2.1

A. Rosenfeld and J. L. Pfaltz. Sequential operations in digital picture processing. *J. ACM*, 13 (4):471–494, Oct. 1966. ISSN 0004-5411. doi: 10.1145/321356.321357. URL https://doi.org/10.1145/321356.321357. 3.1.4, 6.4.1, D

B. T. P. Rowe, M. Jarvis, R. Mandelbaum, G. M. Bernstein, J. Bosch, M. Simet, J. E. Meyers, T. Kacprzak, R. Nakajima, J. Zuntz, H. Miyatake, J. P. Dietrich, R. Armstrong, P. Melchior, and M. S. S. Gill. GALSIM: The modular galaxy image simulation toolkit. *Astronomy and Computing*, 10:121–150, Apr. 2015. doi: 10.1016/j.ascom.2015.02.002. 6.5.1

R. Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1(2):127–190, 1999. 4.2.4

S. Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 4.2.1, 4.2.7

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985. 4.2.2, D

D. E. Rumelhart, G. E. Hinton, R. J. Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988. 4.2.2, D

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 4.3.3

T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017. 6.3

J. Sánchez and F. Perronnin. High-dimensional signature compression for large-scale image classification. In *CVPR 2011*, pages 1665–1672. IEEE, 2011. 4.3.3

S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry. How does batch normalization help optimization? In *Advances in Neural Information Processing Systems*, pages 2483–2493, 2018. 4.2.7

P. L. Schechter, M. Mateo, and A. Saha. DoPHOT, A CCD Photometry Program: Description and Tests. PASP, 105:1342, Nov. 1993. doi: 10.1086/133316. 1, C

J. Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015. 1

J. Serra. Morphology for grey-tone functions. *Image analysis and mathematical morphology*, pages 424–478, 1982. 3.2

J. Serra. *Image Analysis and Mathematical Morphology: Vol.: 2: Theoretical Advances*. Academic Press, 1988. 3.2

J. L. Sérsic. Influence of the atmospheric and instrumental dispersion on the brightness distribution in a galaxy. *Boletin de la Asociacion Argentina de Astronomia La Plata Argentina*, 6: 41–43, Feb. 1963. 6.5.1

C. Shapiro, E. Huff, and R. Smith. Intra-pixel response characterization of a HgCdTe near infrared detector with a pronounced crosshatch pattern. In *High Energy, Optical, and Infrared Detectors for Astronomy VIII*, volume 10709 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, page 1070936, July 2018. doi: 10.1117/12.2314431. 2.4.5

J. Sietsma and R. J. Dow. Creating artificial neural networks that generalize. *Neural networks*, 4(1):67–79, 1991. 4.2.5

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 4.3.3, 5.3.1, 5.5.1, 6.4.4, A.2, C, D

J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Deylon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky. *Nonlinear black-box modeling in system identification: a unified overview*. Linköping University, 1995. 4.2.5, 4.12, 4.2.5

E. Slezak, A. Bijaoui, and G. Mars. Galaxy counts in the Coma supercluster field. II. Automated image detection and classification. A&A, 201:9–20, July 1988. 3.1.1, 3.1.2, 3.1.4, 3.1.5

E. Slezak, A. Bijaoui, and G. Mars. Identification of structures from galaxy counts : use of the wavelet transform. A&A, 227:301–316, Jan. 1990. 3.3.2

E. Slezak, F. Durret, and D. Gerbal. A Wavelet Analysis Search for Substructures in Eleven X-Ray Clusters of Galaxies. AJ, 108:1996, Dec. 1994. doi: 10.1086/117212. 3.3.2

S. A. Solla, E. Levin, and M. Fleisher. Accelerated learning in layered neural networks. *Complex Systems*, 2, 1988. 4.2.4

L. R. Spitler, F. D. Longbottom, J. A. Alvarado-Montes, A. E. Bazkiaei, S. E. Caddy, W. T. Gee, A. Horton, S. Lee, and D. J. Prole. The Huntsman Telescope. *arXiv e-prints*, art. arXiv:1911.11579, Nov. 2019. 1, C

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15 (1):1929–1958, 2014. 4.2.5

J.-L. Starck and J. Bobin. Astronomical Data Analysis and Sparsity: from Wavelets to Compressed Sensing. *arXiv e-prints*, art. arXiv:0903.3383, Mar. 2009. 3.3.3

J. L. Starck and M. Pierre. Structure Detection in Low Intensity X-Ray Images using the Wavelet Transfor M Applied to Galaxy Cluster Cores Analysis. In R. Albrecht, R. N. Hook, and H. A. Bushouse, editors, *Astronomical Data Analysis Software and Systems VII*, volume 145 of *Astronomical Society of the Pacific Conference Series*, page 500, Jan. 1998. 3.3.2

J. L. Starck, H. Aussel, D. Elbaz, D. Fadda, and C. Cesarsky. Faint source detection in ISOCAM images. A&AS, 138:365–379, Aug. 1999. doi: 10.1051/aas:1999281. 3.3

J. L. Starck, A. Bijaoui, I. Valtchanov, and F. Murtagh. A combined approach for object detection and deconvolution. A&AS, 147:139–149, Nov. 2000. doi: 10.1051/aas:2000293. 1, C

J. L. Starck, H. Aussel, D. Elbaz, D. Fadda, and R. Gastaud. PRETI: ISOCAM Faint Sources Detection. In L. Metcalfe, A. Salama, S. B. Peschke, and M. F. Kessler, editors, *The Calibration Legacy of the ISO Mission*, volume 481 of *ESA Special Publication*, page 451, Jan. 2003. 3.3.2, D

P. B. Stetson. DAOPHOT: A Computer Program for Crowded-Field Stellar Photometry. PASP, 99:191, Mar. 1987. doi: 10.1086/131977. 1, 3.1.1, 3.1.2, 3.1.3, 3.1.5, 3.1.6, C

A. J. Storkey, N. C. Hambly, C. K. I. Williams, and R. G. Mann. Cleaning sky survey data bases using Hough transform and renewal string approaches. MNRAS, 347(1):36–51, Jan. 2004. doi: 10.1111/j.1365-2966.2004.07211.x. 5.2

E. Suchyta, E. M. Huff, J. Aleksić, P. Melchior, S. Jouvel, N. MacCrann, A. J. Ross, M. Crocce, E. Gaztanaga, K. Honscheid, B. Leistedt, H. V. Peiris, E. S. Rykoff, E. Sheldon, T. Abbott, F. B. Abdalla, S. Allam, M. Banerji, A. Benoit-Lévy, E. Bertin, D. Brooks, D. L. Burke, A. Carnero Rosell, M. Carrasco Kind, J. Carretero, C. E. Cunha, C. B. D'Andrea, L. N. da Costa, D. L. DePoy, S. Desai, H. T. Diehl, J. P. Dietrich, P. Doel, T. F. Eifler, J. Estrada, A. E. Evrard, B. Flaugher, P. Fosalba, J. Frieman, D. W. Gerdes, D. Gruen, R. A. Gruendl, D. J. James, M. Jarvis, K. Kuehn, N. Kuropatkin, O. Lahav, M. Lima, M. A. G. Maia, M. March, J. L. Marshall, C. J. Miller, R. Miquel, E. Neilsen, R. C. Nichol, B. Nord, R. Ogando, W. J. Percival, K. Reil, A. Roodman, M. Sako, E. Sanchez, V. Scarpine, I. Sevilla-Noarbe, R. C. Smith, M. Soares-Santos, F. Sobreira, M. E. C. Swanson, G. Tarle, J. Thaler, D. Thomas, V. Vikram, A. R. Walker, R. H. Wechsler, Y. Zhang, and DES Collaboration. No galaxy left behind: accurate measurements with the faintest objects in the Dark Energy Survey. MNRAS, 457(1):786–808, Mar. 2016. doi: 10.1093/mnras/stv2953. 1, C

S. Sukittanon, A. C. Surendran, J. C. Platt, and C. J. Burges. Convolutional networks for speech detection. In *Eighth international conference on spoken language processing*, 2004. 4.3.2

A. S. Szalay, A. J. Connolly, and G. P. Szokoly. Simultaneous Multicolor Detection of Faint Galaxies in the Hubble Deep Field. AJ, 117(1):68–74, Jan. 1999. doi: 10.1086/300689. 3.1.1, 3.1.1, 3.1.4

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 4.2.5

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 4.3.3, A.1, A.2

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 4.2.5

V. I. Tatarskii. *Wave Propagation in Turbulent Medium*. 1961. 2.4.3

L. Taylor and G. Nitschke. Improving deep learning using generic data augmentation. *arXiv preprint arXiv:1708.06020*, 2017. 4.2.5

C.-H. Teh and R. T. Chin. On image analysis by the methods of moments. *IEEE Transactions on pattern analysis and machine intelligence*, 10(4):496–513, 1988. 4.2.7

H. Teimoorinia, J. J. Kavelaars, S. D. J. Gwyn, D. Durand , K. Rolston, and A. Ouellette. Assessment of Astronomical Images Using Combined Machine-learning Models. AJ, 159(4): 170, Apr. 2020. doi: 10.3847/1538-3881/ab7938. 5.3.2

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. 4.2.5

L. Tortorelli, M. Fagioli, J. Herbel, A. Amara, T. Kacprzak, and A. Refregier. Measurement of the B-band Galaxy Luminosity Function with Approximate Bayesian Computation. *arXiv e-prints*, art. arXiv:2001.07727, Jan. 2020. 1, C

T. Tran, T. Pham, G. Carneiro, L. Palmer, and I. Reid. A bayesian data augmentation approach for learning deep models. In *Advances in neural information processing systems*, pages 2797–2806, 2017. 4.2.5

G. Turin. An introduction to matched filters. *IRE transactions on Information theory*, 6(3): 311–329, 1960. 3.1.2, D

M. Unser and A. Aldroubi. Polynomial splines and wavelets—a signal processing perspective. *Wavelets: a tutorial in theory and applications*, pages 91–122, 1992. 3.3.2

A. Vafaei Sadr, E. E. Vos, B. A. Bassett, Z. Hosenie, N. Oozeer, and M. Lochner. DEEPSOURCE: point source detection using deep learning. MNRAS, 484(2):2793–2806, Apr. 2019. doi: 10.1093/mnras/stz131. 6.3

F. Valdes, R. Gruendl, and DES Project. The DECam Community Pipeline. In N. Manset and P. Forshay, editors, *Astronomical Data Analysis Software and Systems XXIII*, volume 485 of *Astronomical Society of the Pacific Conference Series*, page 379, May 2014. 5.4.1

G. C. Valley and S. M. Wandzura. Spatial correlation of phase-expansion coefficients for propagation through atmospheric turbulence. *Journal of the Optical Society of America (1917-1983)*, 69(5):712, May 1979. 2.4.3

D. Valls-Gabaud and MESSIER Collaboration. The MESSIER surveyor: unveiling the ultra-low surface brightness universe. In A. Gil de Paz, J. H. Knapen, and J. C. Lee, editors, *Formation and Evolution of Galaxy Outskirts*, volume 321 of *IAU Symposium*, pages 199–201, Mar. 2017. doi: 10.1017/S1743921316011388. 1, C

P. G. van Dokkum. Cosmic-Ray Rejection by Laplacian Edge Detection. PASP, 113:1420–1427, Nov. 2001. doi: 10.1086/323894. 5.2, 5.4.1, 5.6.1, 5.6.1, D

P. G. van Dokkum, J. Bloom, and M. Tewes. L.A.Cosmic: Laplacian Cosmic Ray Identification, July 2012. 5.2

B. Vandame. New algorithms and technologies for the un-supervised reduction of Optical/IR images. In J.-L. Starck and F. D. Murtagh, editors, *Astronomical Data Analysis II*, volume 4847 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 123–134, Dec. 2002. doi: 10.1117/12.460591. 5.2, 5.4.1

A. Vikhlinin, W. Forman, C. Jones, and S. Murray. ROSAT Extended Medium-Deep Sensitivity Survey: Average Source Spectra. ApJ, 451:564, Oct. 1995. doi: 10.1086/176244. 3.1.1, 3.1.2, 3.1.3

J. Wang and L. Perez. The effectiveness of data augmentation in image classification using deep learning. *Convolutional Neural Networks Vis. Recognit*, page 11, 2017. 4.2.5

A. S. Weigend, B. A. Huberman, and D. E. Rumelhart. Predicting the future: A connectionist approach. *International journal of neural systems*, 1(03):193–209, 1990. 4.2.5, 4.2.5

D. C. Wells, E. W. Greisen, and R. H. Harten. FITS - a Flexible Image Transport System. A&AS, 44:363, June 1981. 2.5

C. K. Williams. Prediction with gaussian processes: From linear regression to linear prediction and beyond. In *Learning in graphical models*, pages 599–621. Springer, 1998. 5.4.1

R. N. Wilson. *Reflecting telescope optics. Part 1: Basic design theory and its historical development*. 2000. 2.4

R. N. Wilson. *Reflecting telescope optics II : manufacture, testing, alignment modern techniques*. 2001. 2.4

R. A. Windhorst, N. P. Hathi, S. H. Cohen, R. A. Jansen, D. Kawata, S. P. Driver, and B. Gibson. High resolution science with high redshift galaxies. *Advances in Space Research*, 41(12):1965–1971, Jan. 2008. doi: 10.1016/j.asr.2007.07.005. 6.7

S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987. 4.2.7

T. Wolfe, T. Armandroff, M. M. Blouke, T. Rector, R. Reed, A. Saha, R. Schommer, C. Smith, R. M. Smith, and A. R. Walker. CCD detector performance for NOAO's wide-field MOSAIC cameras. In M. M. Blouke, N. Sampat, G. M. Williams, and T. Yeh, editors, Proc. SPIE, volume 3965 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 80–91, May 2000. doi: 10.1117/12.385427. 5.1

B. K. Wong, T. A. Bodnovich, and Y. Selvi. Neural network applications in business: A review and analysis of the literature (1988–1995). *Decision Support Systems*, 19(4):301 – 320, 1997. ISSN 0167-9236. doi: https://doi.org/10.1016/S0167-9236(96)00070-X. URL http://www.sciencedirect.com/science/article/pii/S016792369600070X. 4.2.7

P. Woodward. Probability and information theory, with applications to radar. new york: Mcraw-hill book co. inc, 1953. 1, 3.1.2, C, D

P. M. Woodward. *Probability and Information Theory, with Applications to Radar: International Series of Monographs on Electronics and Instrumentation*, volume 3. Elsevier, 2014. 1, 3.1.2, C, D

S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 4.3.3, A.1

J. Xu, A. G. Schwing, and R. Urtasun. Tell me what you see and i will show you where it is. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3190–3197, 2014. 5.5.2

L. Xu, A. Krzyzak, and C. Y. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE transactions on systems, man, and cybernetics*, 22 (3):418–435, 1992. 4.2.7

L. S. Yaeger, R. F. Lyon, and B. J. Webb. Effective training of a neural network character classifier for word recognition. In *Advances in neural information processing systems*, pages 807–816, 1997. 4.2.5

T. Yang, Y. Wu, J. Zhao, and L. Guan. Semantic segmentation via highly fused convolutional network with multiple soft cost functions. *Cognitive Systems Research*, 2018. 5.5.1, 5.5.2

Y. Yang, N. Li, and Y. Zhang. Automatic moving object detecting and tracking from astronomical ccd image sequences. In *2008 IEEE International Conference on Systems, Man and Cybernetics*, pages 650–655. IEEE, 2008. 3.2.2

H. K. C. Yee. A Faint-Galaxy Photometry and Image-Analysis System. PASP, 103:396, Apr. 1991. doi: 10.1086/132834. 3.1.1, 3.1.3

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 5.3.1

M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *2011 International Conference on Computer Vision*, pages 2018–2025. IEEE, 2011. 5.3.1

K. Zhang and J. S. Bloom. deepCR: Cosmic Ray Rejection with Deep Learning. ApJ, 889(1): 24, Jan. 2020. doi: 10.3847/1538-4357/ab3fa6. 5.3.2

M. Zhang and J. Kainulainen. Deep point spread function photometric catalog of the VVV survey data. A&A, 632:A85, Dec. 2019. doi: 10.1051/0004-6361/201935513. 1, C

C. Zheng, J. Pulido, P. Thorman, and B. Hamann. An improved method for object detection in astronomical images. MNRAS, 451(4):4445–4459, Aug. 2015. doi: 10.1093/mnras/stv1237. 3.1.5, 3.2.2

Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*, 2017. 4.2.5

X. Zhou, D. Wang, and P. Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 6.2.3

J. Zoubian, M. Kümmel, S. Kermiche, N. Apostolakos, A. Chapon, A. Ealet, P. Franzetti, B. Garilli, E. Jullo, and L. Paioro. Instrument Simulations of the EUCLID/NISP Spectrometer. In N. Manset and P. Forshay, editors, *Astronomical Data Analysis Software and Systems XXIII*, volume 485 of *Astronomical Society of the Pacific Conference Series*, page 509, May 2014. 5.6.1