UNIVERSITE GUSTAVE EIFFEL

ECOLE DOCTORALE MSTIC

Discipline : Informatique

Object Detection using Component-Graphs and ConvNets with Application to Astronomical Images

---

Thanh Xuan NGUYEN

10/09/2021

Soutenue publiquement à ESIEE Paris devant le jury composé de:

Benoît NAEGEL, University of Strasbourg	Rapporteur
Sébastien LEFEVRE, University of South Brittany	Rapporteur
Odyssée MERVEILLE, INSA Lyon	Rapporteur
Marc HUERTAS-COMPANY, University of Paris	Rapporteur
Giovanni CHIERCHIA, Université Gustave Eiffel	Co-encadrant
Benjamin PERRET, Université Gustave Eiffel	Co-encadrant
Hugues TALBOT, Université Paris-Saclay	Directeur de thèse
Laurent NAJMAN, Université Gustave Eiffel	Directeur de thèse

Acknowledgement

First and foremost, I would like to acknowledge my supervisors Benjamin Perret, Giovanni Chierchia, Hugues Talbot, and Laurent Najman, for their invaluable support and guidance along my Ph.D. journey. I would also like to thank my external supervisors/mentors Reynier Peletier, Michael H.F. Wilkinson, Peter Tino, and Mathieu Aubry for your advice and feedback.

My deepest gratitude is due to the jury members, Benoît Naegel, Sébastien Lefèvre, Odyssée Merveille, Marc Huertas-Company, for their time and comments. It was a great honour for me to present my work to them.

This work was fully funded by the Marie Skłodowska-Curie grant and by the Programme d'Investissements d'Avenir. Beyond financial aspects, they are generous and open-minded networking. I was able to connect, learn, and coordinate with worldwide researchers, specifically with the ESR fellow friends Alan, Caroline, Aleke, Teymoor, Mohamad, Abolfazl, Marco, Alex, Angela, Bahar, Nushkia, Shivangee, and Michele.

I would like to acknowledge LIGM Lab - ESIEE Paris - Université Gustave Eiffel for the top-tier research environment. I was lucky to be there and to meet many scientists and friends, Yukiko, Jean, Lama, Vincent, Yamna, Sarah, Clara, Monika, Deise, Gia Thuy, Gabriel, Caroline, Stéphane, Bruno, Rosemberg, Diane, Kacper, Elias, Edward, Karla, Sami, Yuki, Jordao, Tanya, Yasin, Ahmed, and Sofien. I would like to thank you all for your kindness, empathy, and advice.

Thanks to my very first former supervisor Thanh Ha Le who has always encouraged me to persuade and discover science. Thanks to my dear friends Hung, Tung, Hiep, and Erick for always being my friends.

Finally, to my lovely Hong Van, my little Ngoc Thu, and my dear family: thank you all, I would not have been strong enough to go through this journey without you!

Abstract

In this work, we investigate object detection algorithms with application to astronomical images [*]. We specifically target to detect faint astronomical sources which are near the image background level. Our main directions include Mathematical Morphology (MM) and Convolutional Neural Network (ConvNet). The contributions of this study are presented in two parts:

The first part proposes a novel morphological-based approach based on component-graphs and statistical hypothesis tests. The component-graphs can efficiently handle multi-band images while the statistical hypothesis tests can identify components that are significantly different from the background level. Beyond the classical component-trees and their multivariate extensions, the component-graph holds the complete structural information of multi-band images as directed acyclic graphs (DAGs). Such DAGs are more general and more powerful at the cost of non-trivial object filtering algorithms. Then, we introduce two algorithms to filter duplicated and partial components in the component-graphs. Experiments demonstrate that our proposed approach significantly improves object detection on both multi-band simulated and real astronomical images.

The second part turns our attention to ConvNet direction. We introduce a real dataset of annotated astronomical objects. Based on this dataset, we propose two models: a ConvNet-based model and a hybrid model. The ConvNet-based model tailors astronomical contexts with three novel components, including a normalization layer, an object differentiation module, and a smoothness regularizer. Besides, the hybrid model uses both Morphology and ConvNet. In the hybrid method, morphological modules select region proposals while ConvNet extracts relevant information from the selected pro-

[*]This research was funded by the European Union Horizon 2020 program under the Marie Skłodowska-Curie grant agreement No. 721463 to the SUNDIAL ITN network and by the Programme d'Investissements d'Avenir (LabEx BEZOUT ANR-10-LABX-58).

posals. Ablation studies show that the two proposed models outperform the state of the art on both synthetic and real datasets.

Keywords Object Detection, Astronomical Images, Mathematical Morphology, Component-graphs, ConvNet, R-CNN.

Résumé Long

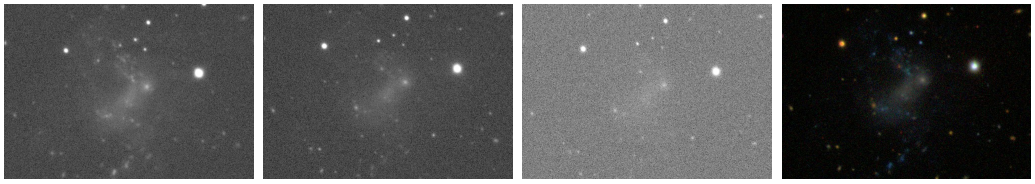
Resumé

This thesis aims at developing efficient object detection algorithms with applications to astronomical images. We have explored the use of Mathematical Morphology (MM) and Convolutional Neural Network (ConvNet) in our proposed models. In astronomy, object detection (or finding sources) is the fundamental preliminary stage before entering any analysis. Despite the long historical development of astronomical source finders, it is challenging to detect faint sources and to segment crowded sources. Faint structures stand for structures lying near background levels while crowded sources are structures at interacting regions. To tackle these difficulties, we relies on three main ideas: *Component-graphs*, *ConvNets*, and *Astronomical Context*. We have proposed three models, including a morphological-based model, a ConvNet-based model, and a hybrid model. Experiments and ablation studies demonstrate our proposed models gain significant improvements in detecting objects on both multi-band simulated and real astronomical datasets.

A. Object Detection in Astronomy

We first cover the basis of astronomical images and the challenges of object detection (or finding sources) in astronomy. We discuss our research interests and methodological directions that lead to a novel dataset (in Sec. B) and our proposed MM-based (in Sec. C) and ConvNet-based (in Sec. D) approaches.

In astronomy, astronomers measure the source's radiation with an optical filter produces a *single-band image*. The optical filter lets a certain wavelength interval pass through. Normally, several images of the same field of view are obtained with different filters covering several standard wavelength



(a) Filter g (b) Filter r (c) Filter z (d) Filters (g,r,z)

Figure 1: The UGC 07332: (a-c) Three single-band images associate to three filters g r z and (d) The composition of the g,r,z images, source SDSS.

ranges of interest. A *multi-band image* is the stack of these same-of-view single-band images. Fig. 1 shows similar filters of UGC 07332 - a nearby, blue, low surface brightness galaxy. As we can see, the multi-band image provides a determination of the image colors.

To design efficient astronomical object detection models, we relies on three main ideas: *Component-graphs*, *ConvNet*, and *Astronomical Context*.

- **Component-graphs**: The information gain of the multi-band images is useful to improve both object detection and segmentation, see Fig. 1. For object detection, the information gain gives us more confident at detecting faint structure that lies near the background level. For object segmentation, the color information of the multi-band images helps to deblend interacting regions.

  To take advantage of the multi-band images, we propose to use *component-graph* structures in our model, see Chapter 3. Compared to classical component-trees, such component-graphs [Passat and Naegel, 2014] are more general and more powerful at the cost of higher construction and filtering complexities.

- **ConvNet:** We have chosen to integrate *ConvNet* into our models to improve both object detection and segmentation, see Chapter 5. The ConvNet architectures can naturally process multi-band images.

  In contrast to morphology, ConvNet does not limit segmentation masks to the thresholded components, then we have some degree of freedom to define and optimize CNN-based models that allow overlapping segmentation.

- **Astronomical Context**: Astronomical images are very different from

natural images in terms of range, quantization, size, and other characteristics. We see that many existing source finders just apply computer-vision models without considering these differences. We target to tailor the base models with characteristics of the astronomical context.

In astronomy, we observe that the center of the sources is usually brighter and better localized than the outer parts, i.e., the center is more important than the outer parts. We name it *Centralization characteristic*. We have used that observation in several elements in our proposed models. In Chapter 3, the centralization characteristic is used to differentiate duplicated components in the component-graph. In Chapter 5, the same characteristic is used in CC-NMS module to detect multiple detections and being used in a smoothness regularizer. Also, the difference between astronomical and natural images motivates the development of a normalization layer in Chapter 5.

B. Astronomical Datasets

This work has used both simulated and real multi-band astronomical images. In addition to the *FDS Simulation* [Venhola, 2019], we introduce a *Real KiDS Dataset* of multi-band astronomical objects. The objects on the real KiDS images are annotated semi-automatically, as shown in Fig. 2.

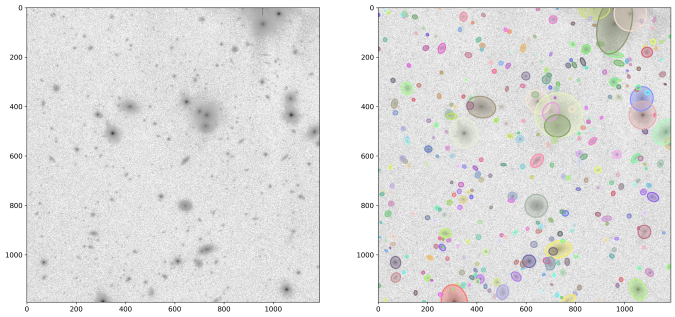


Figure 2: Real dataset: The KiDS images (left) and annotations (right).

The idea is to use high-quality reference images to manually correct automated detections on lower-quality images. The lower-quality images are the real KiDS images, while the references are the images sharing the same

field of view taken from the Hubble Space Telescope Cosmic Assembly Near-infrared Deep Extra-galactic Legacy Survey [Hubble, 2000] (HST). Since the HST images have a much higher resolution and signal-to-noise ratio than the KiDS images, we can correct the pre-annotated objects with more confidence.

Given the KiDS images and the reference HST images, objects are firstly extracted from the KiDS images using existing automated source finders, such as Sourcerer and MTObject [Teeninga et al., 2016; Wilkinson et al., 2019]. Second, the pre-annotated objects enter a manual correction supported by higher quality HST images.

C. Morphological-based Approach

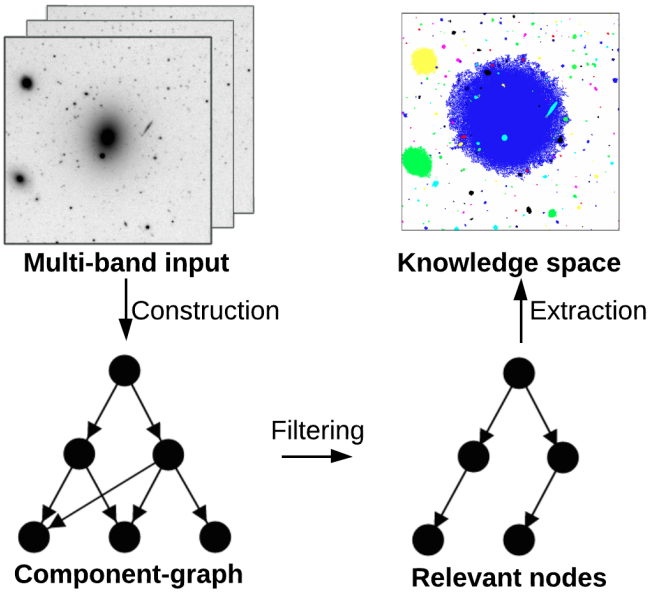


Figure 3: CGO filtering method using component-graphs.

This chapter proposes CGO [Nguyen et al., 2021a, 2020a] - a novel morphological model for object detection in multi-band images, as shown in Fig. 3. The model relies on *component-graphs* and *statistical hypothesis tests*. The component-graph structure holds the whole structural information of multi-band images at the cost of higher construction and filtering complexities. Such information can improve object detection sensitivity and object

segmentation capacity.

The main contributions of this morphological approach include:

- Propose a novel multi-band object detection framework relying on component-graphs and application to astronomical source detection.
- Address that the component-graph is better at capturing image structures comparing to classical component-trees.
- Introduce two filtering algorithms to detect duplicated and partial nodes in the component-graphs.
- Improve object detection results on simulated and real multi-band astronomical images.

Experiments demonstrate a significant improvement in detecting objects on both multi-band simulated and real astronomical images.

D. ConvNet-based Approaches

We propose two models: an *R-CNN-based model* and a *hybrid model* that takes the advantages of both morphological-based and ConvNet-based models to adapt to astronomical contexts, as shown in Fig. 4. On the one hand, ConvNet has shown excellent results in visual perception tasks as convolutional operators can efficiently process multi-band images. On the other hand, ConvNet does not limit segmentation masks to the thresholded components, then we have some degree of freedom to define and optimize models that allow overlapping segmentation.

The main contributions of this approach include:

- We proposed an RCNN-based model tailoring object detection on astronomical images. The novelties of the proposed model consist of: a trainable normalization layer that can be trained end-to-end with the whole model; CC-NMS module is designed to replace the default NMS at removing multiple detections of a single object; and a smoothness regularizer for the segmentation head in the model.
- We discussed a hybrid approach using both morphological trees and R-CNN models for object detection. Intuitively, the hybrid model takes advantage of a morphological tree to detect potential regions in the first

stage, then using convolutional heads to predict relevant information such as labels and segmentation masks.

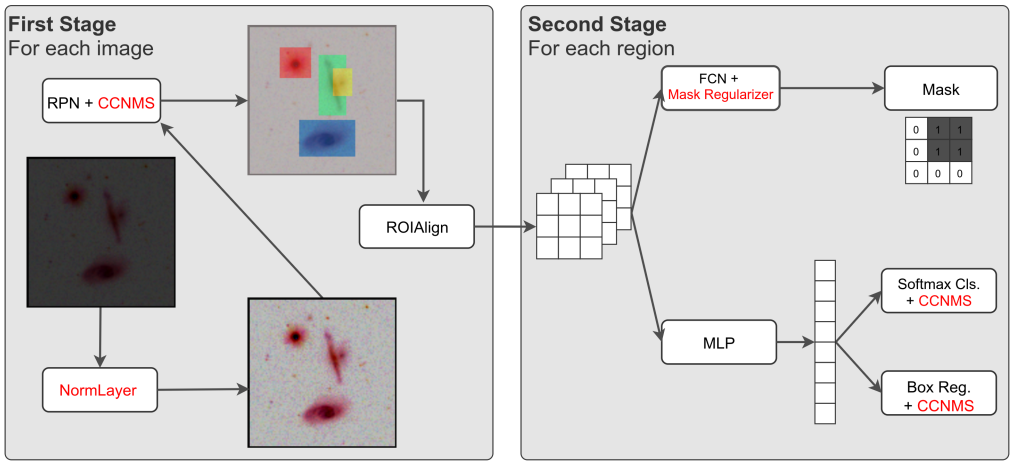


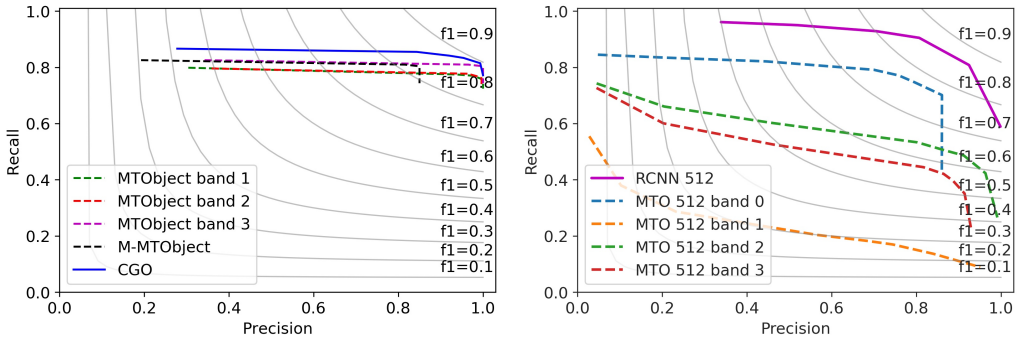
Figure 4: The proposed R-CNN model for astronomical object detection: Three novel modules, including a NormLayer, a CC-NMS module, and a Mask Regularizer are red-highlighted.

E. Experiments and Conclusion

We use precision, recall, and F1-score, as in [Haigh et al., 2020]. The evaluation matches at most one detected object in the detection map to each target object in the ground-truth map. Each target object in the ground-truth map is represented by its brightest pixel called its *representative pixel*, hence each representative pixel is included in at most one object in the detection map. If a detected object contains several representative pixels of different target objects, then the detected object is associated to the target object with the brightest representative pixel.

Comprehensive experiments demonstrate our proposed models can detect astronomical objects on both multi-band simulated and real astronomical images, with significantly better precision and recall than the state-of-the-art method [Haigh et al., 2020] [Teeninga et al., 2016].

In summary, we have proposed three models for astronomical object detection: the morphological-based model - CGO, the ConvNet-based model,



(a) Evaluation on the FDS Simulation. (b) RCNN vs. MTO on KiDS dataset.

Figure 5: Experimental results.

and the hybrid model. Experiments and ablation studies have demonstrated significant improvement of the three proposed models compared to the baseline MTObject. Despite showing great detection performance, a current limitation of the proposed CGO is its time complexity which prevents the processing of large images at once. On the other hand, both ConvNet-based and hybrid models require training datasets that remain less practical for astronomers.

Contents

INTRODUCTION

This thesis aims at developing efficient object detection algorithms with applications to astronomical images. We have explored the use of mathematical morphology (MM) and convolutional neural network (ConvNet) in our proposed models.

In astronomy, object detection (or finding sources) is the fundamental preliminary stage before entering any analysis. Despite the long historical development of astronomical source finders, it is challenging to detect faint sources and to segment crowded sources. Faint structures stand for structures lying near background levels while crowded sources are structures at interacting regions. To tackle these difficulties, we relies on three main ideas: *Component-graphs*, *ConvNets*, and *Astronomical Context*. We have proposed three models, including a morphological-based model, a ConvNet-based model, and a hybrid model. In the following, we summarize the thesis organized in two parts, includes five chapters:

Chapter 1 - Object Detection in Astronomy This chapter introduces and explains the chosen methodological directions of this manuscript for multi-band object detection in astronomy. We first cover the basis of astronomical images and the challenges of astronomical object detection (or finding sources). Then we review and address the pros and cons of existing state-of-the-art source finders. Based on the review, we discuss our research interests and methodological directions that lead to two main proposed approaches presented in the two following parts.

PART I - MATHEMATICAL MORPHOLOGY

In the first part, we develop a morphological-based model to take advantage of multi-band astronomical images. The topic is organized as two chapters:

Chapter 2 - Morphological Connected Operators In this chapter, we present an overview of *Connected Operators* in mathematical morphology that is the main context of our proposed morphological approach for astronomical object detection. First, we cover the historical development of connected operators from the early stage on binary images to the extensions on grey-scale and multi-band images. The review includes primary morphological structures, construction algorithms, and filtering strategies of these structures. Besides, we explicitly focus on several advances of connected operators to handle multi-band images, including Component-graphs, Multivariate Tree-of-Shape, and Connected Component-Tree.

Chapter 3 - Object Detection with Component-graphs This chapter proposes a novel morphological model for object detection in multi-band images. The model relies on *component-graphs* and *statistical hypothesis tests*. The component-graph structure holds the whole structural information of multi-band images at the cost of higher construction and filtering complexities. Such information can improve object detection sensitivity and object segmentation capacity. We first analyze the component-graph capacity at capturing image structures comparing to the classical component-trees. We then introduce two algorithms to filter duplicated and partial nodes in the component-graphs. Experiments demonstrate a significant improvement in detecting objects on both multi-band simulated and real astronomical images.

PART II - COMBINING MORPHOLOGY & CONVNET

The second part turns our attention to ConvNet-based direction to address astronomical object detection. We explore the use of Region-Based Convolutional Neural Network (R-CNN) to tackle object detection in multi-band astronomical images. The topic is organized as two chapters:

Chapter 4 - ConvNet Object Detection Literature

This chapter provides an overview of Convolutional Neural Network-based (ConvNet/CNN) models for visual perception tasks in the field of machine learning. We have chosen to specifically narrow down to the class of Region-based Convolutional Neural Networks (R-CNN) as based model to later develop our ideas. In particular, we describe the generalization, the evolution, and the essential components of the R-CNN variants.

Chapter 5 - Combining ConvNet and Morphology

We propose two models: an *R-CNN-based model* and a *hybrid model* that takes the advantages of both morphological-based and ConvNet-based models to adapt to astronomical contexts. On the one hand, ConvNet has shown excellent results in visual perception tasks as convolutional operators can efficiently process multi-band images. On the other hand, ConvNet does not limit segmentation masks to the thresholded components, then we have some degree of freedom to define and optimize models that allow overlapping segmentation. It is important to note that we introduce a pipeline to acquires a *novel real dataset* of multi-band astronomical images. Then, a series of experiments and ablation studies demonstrate our proposed models gain significant improvements in detecting objects on both multi-band simulated and real astronomical images.

PUBLICATIONS

The results presented in this manuscript have been partially published the following articles:

Journal Articles

- Nguyen, T., Chierchia, G., Razim, O., Peletier, R., Najman, L., Talbot, H., and Perret, B. (2021a). Object detection with component-graphs in multi-band images: Application to source detection in astronomical images. *IEEE Access*, pages 156482–15649

Conference Proceedings

- Nguyen, T. X., Chierchia, G., Najman, L., Venhola, A., Haigh, C., Peletier, R., Wilkinson, M. H. F., Talbot, H., and Perret, B. (2020a). Cgo: Multiband astronomical source detection with component-graphs. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 16–20. IEEE

- Wilkinson, M. H. F., Haigh, C., Gazagnes, S., Teeninga, P., Chamba, N., Nguyen, T. X., Talbot, H., Najman, L., Perret, B., Chierchia, G., Venhola, A., and Peletier, R. (2019). Sourcerer: A robust, multi-scale source extraction tool suitable for faint and diffuse objects. In *IAU 355 Symposium*

Talks and Workshops

- Nguyen, T. X., Chierchia, G., Talbot, H., Najman, L., and Perret, B. (2021c). Astronomical object detection with morphology and deep learning. *UGE Atelier Doctorant* [Talk]

- Nguyen, T. X., Chierchia, G., Talbot, H., Najman, L., and Perret, B. (2020b). Astronomical source detection with deep learning. *Faint Object Detection SUNDIAL* [Talk]

- Nguyen, T. X., Chierchia, G., Talbot, H., Najman, L., and Perret, B. (2020c). Cgo: Multiband astronomical source detection with component-graphs. *Journée ISS France* [Talk]

Chapter 1

Object Detection in Astronomy

This chapter aims at explaining the chosen methodological directions of this manuscript for multi-band object detection in astronomy. We start by giving the basis of astronomical images and the challenges of astronomical object detection (or finding sources) in Sec. 1.1. Then Sec. 1.2 reviews and addresses the pros and cons of existing state-of-the-art source finders. Finally, Sec. 1.3 discusses our research interests and directions for object detection in this work.

Contents

1.1 Astronomical Context

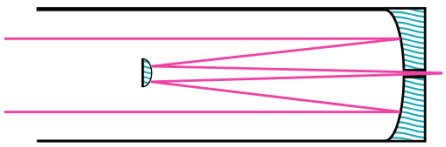
The main references for this section are the two books *Electronic Imaging in Astronomy* by [McLean, 2008] and *Astronomy A Physical Perspective* by [Kutner, 2003]. The astronomical image acquisition process is reviewed in Sec. 1.1.1 while Sec. 1.1.2 focuses on single-band and multi-band images.

1.1.1 Image Acquisition in Astronomy

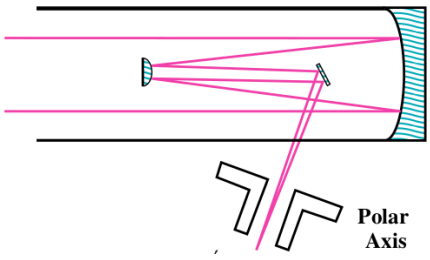
What constitutes a perfect image acquisition system in astronomy? This section briefly answers the question by covering the main aspects for astronomical image acquisition, ranging from capturing devices (telescope design and Charge-Coupled Devices CCDs) to environmental effects (atmosphere and the Point Spread Function PSF).

Telescope

(a) Newtonian



(b) Cassegrain



(c) Coudé

Figure 1.1: Focal arrangements in (a) Newtonian, (b) Cassegrain and (c) Coudé telescopes. In each case the light enters the telescope from left to right [Kutner, 2003].

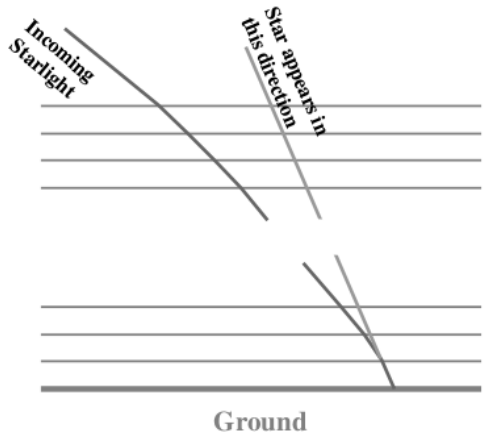


Figure 1.2: Bending of a light ray as it passes through the atmosphere with different fraction layers [McLean, 2008].

The first and foremost element of any imaging system in astronomy is the telescope. The telescopes can be thought as of a camera system with a special design to capture distant objects that we can not see with naked eyes.

The telescope system generally consists of two or three mirrors: a *Primary Mirror* collects a maximum of incoming light; a *Secondary Mirror* focuses the flux; and an optional third plane mirror redirects the rays towards the sensor. Fig. 1.1 illustrates the basic layouts of the Newtonian, the Cassegrain, and the Coudé telescopes.

We can think of light as a stream of photons coming from the space objects to the receiver (telescope or eye) with a certain number of photons per unit area per second. It is trivial that the more photons the receiver collects, the more information the receiver captures. The telescope provides a large collecting area and a long exposure time to intercept as much of the incoming photons as possible. For each frame taken, naked eyes fix exposures time about 1/20 second while modern telescopes can exposure up to hours. In other words, the telescope can see much fainter objects compared to human eyes.

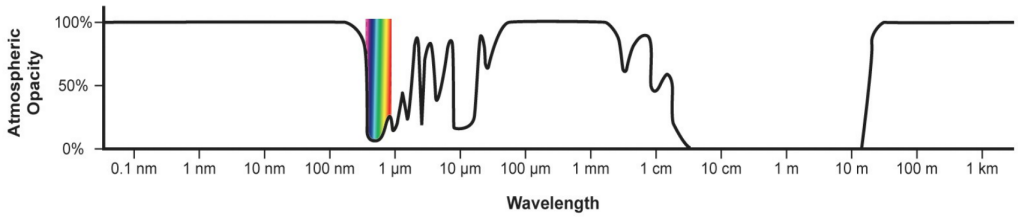


Figure 1.3: Atmospheric absorption percentages throughout the electromagnetic spectrum, source NASA.

Charge-Coupled Devices (CCDs)

The collected light (photons) coming out from the telescope focal system will be recorded by CCDs - the detector system. CCDs contain a grid of high quantum efficiency detectors. Each grid element corresponds to a pixel. The element record the intensity of light (i.e., the number of photons) striking the pixel. To use these CCDs records, there has to be a read-out and data handling phase.

Atmosphere

The light has to pass through the Earth's atmosphere before reaching the telescopes (except space telescopes). The atmosphere absorbs, transmits, and refracts incoming light differently at different wavelengths.

For refraction, the atmosphere can be thought of as multiple layers with different refraction indexes. When light passes from one layer to the other, it is bent towards the vertical direction, as shown in Fig. 1.2. For absorption (how much energy is absorbed) and transmission (how much energy is able to pass through), the atmosphere absorbs/transmits electromagnetic energy at certain wavelengths, see Fig. 1.3. While most of the energy in the Ultraviolet wavelength is absorbed, very little energy in the Visible wavelength is absorbed. In contrast, the Visible wavelength can largely pass through the atmosphere.

If the Earth's atmosphere were stable, the absorption, transmission, and refraction effects would be corrected. However, it varies a lot, causing blurred image, called *the seeing* effect in astronomy. The seeing effect is generally approximated by Point Spread Function (PSF) [McLean, 2008].

A solution to avoid the atmosphere effects is to use space telescopes, such as Hubble Space Telescope. From space-based stations, observation can

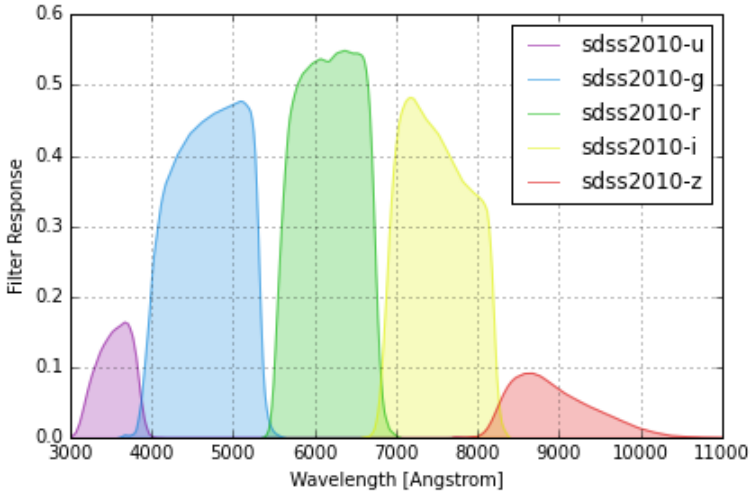


Figure 1.4: SDSS optical filter responses [Doi et al., 2010].

be carried out without atmosphere refraction, absorption, and transmission to acquire higher quality images. However, space telescopes are costly, the majority are currently ground-based telescopes.

1.1.2 Single-band and multi-band Images

Optical Filters We do not measure the source's radiation at a wavelength or at the whole spectrum, we instead measure them in some wavelength ranges. These ranges are defined by *Optical Filters* that let a certain wavelength interval pass through. When using an optical filter, we actually measure the integral of incoming energy over some wavelength range.

Standard filters are U (for ultraviolet), B (for blue), V (for visible, meaning the center of the visible part of the spectrum), R (for red), and I (for infrared). Fig. 1.4 present the five filter curves that have been used in the SDSS Survey [Blanton et al., 2017].

Single-band and Multi-band Images The measurement of the source's radiation with an optical filter produces a *single-band image*. Normally, several images of the same field of view are obtained with different filters covering several wavelength ranges of interest. A *multi-band image* is the stack of these same-of-view single-band images. Fig. 1.5 presents images of the

(a) Filter g

(b) Filter r

(c) Filter i

(d) The composition of filter (g, r, i)

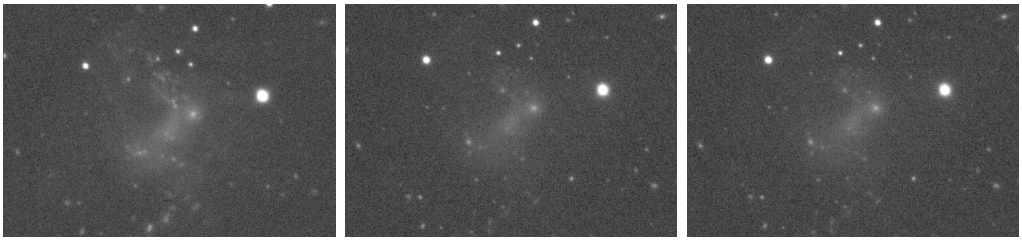
Figure 1.5: The M51 Galaxy: (a-c) Three single-band images associate to three filters g r i and (d) The composition of the three single-band images, source SDSS.

Galaxy M51 in three SDSS filters (g, r, i) and the color composition of them. Fig. 1.6 shows similar filters of UGC 07332 - a nearby, blue, low surface brightness galaxy. As we can see, some details are only visible in the image compositions.

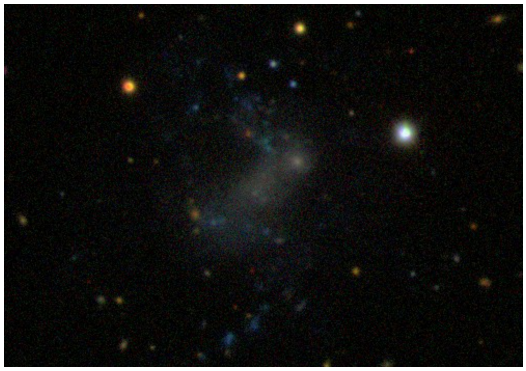
The multiple bands allow a determination of the colors of the image. Technically, the multi-band image acquisition is usually observed simultaneously in one go for all filters to have the same environmental effects.

1.2 Astronomical Object Detection

In astronomy, object detection (or finding sources) is the fundamental preliminary stage before entering any further analysis. The following sections



(a) Filter g (b) Filter r (c) Filter i



(d) The composition of filter (g, r, i)

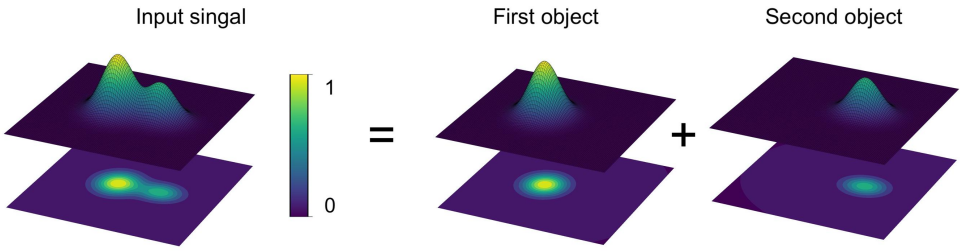
Figure 1.6: The UGC 07332: (a-c) Three single-band images associate to three filters g r i and (d) The composition of the three single-band images, source SDSS.

review the pros and cons of existing methods that mainly fall into two directions using mathematical morphology and convolutional neural network.

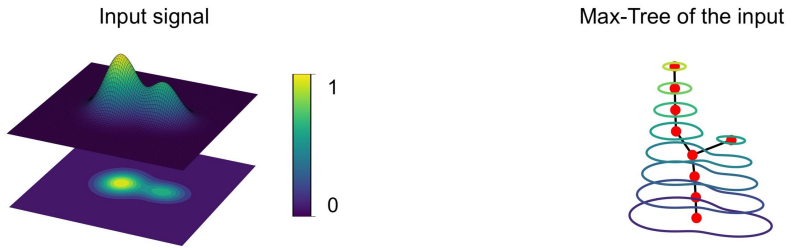
1.2.1 Morphological Approaches and Limitations

SExtractor [Bertin and Arnouts, 1996] is the most widely known and standard use program. The primary strategy of SExtractor relies on filtering a coarse thresholding structure of the input image. It is fast and easy to use but performs poorly at the detection of faint and diffused objects.

To go deeper into the noise, MTObject/Sourcerer [Teeninga et al., 2016] [Wilkinson et al., 2019] suggested filtering a fine thresholding structure of input image, namely Max-Tree, a type of component-trees widely used in mathematical morphology. Thanks to the fine thresholding, the Max-Tree



(a) A simulated single-band image containing two Gaussian-like objects: Pixel intensity is viewed as elevation for visualization purpose.



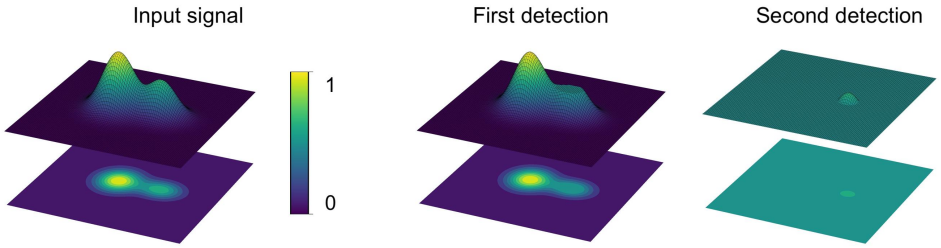
(b) The simulated image and its corresponding Max-Tree.

Figure 1.7: Simulation: a single-band image and its morphological Max-Tree.

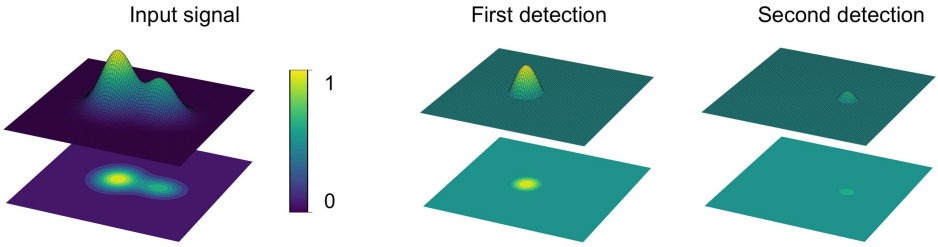
can represent input images as the hierarchy of connected components without losing any bit of information, i.e., the input image can be recovered entirely from the Max-Tree. MTObject/Sourcerer uses statistical hypothesis testing to determine potential connected components which are significantly different from the background level. MTObject/Sourcerer is efficient to detect faint sources while having far fewer parameters SExtractor.

In addition, NoiseChisel [Akhlaghi and Ichikawa, 2015] is another program leveraging mathematical morphology. It iteratively performs binary thresholding to separate background and foreground. Noisechisel claims to be able to detect nebulous objects, it is fast but is designed to be hand-tuned with a lot of parameters.

Last but not least of morphological-based tools, ProFound [Robotham et al., 2018] is a watershed-based method. It firstly estimates a background image and then applies watershed algorithms on the background-subtracted



(a) Over-segmented case: The first object is over-segmented while the second is under-segmented.



(b) Under-segmented case: The both objects are under-segmented.

Figure 1.8: Simulation: usual object segmentation strategies on the Max-Tree representation of the single-band simulated image.

image to produce initial segmentation. An iterative segmentation dilation and background re-estimation is performed to obtain a final segmentation. Practically, ProFound is more suitable for galaxy profiling.

Limitations

The two limitations of existing morphological-based methods [Bertin and Arnouts, 1996; Teeninga et al., 2016; Robotham et al., 2018; Akhlaghi and Ichikawa, 2015; Wilkinson et al., 2019] are *single-band processing* and *de-blending crowded sources*.

- **Single-band Processing.** Even though multi-band astronomical images are available, most source finders were designed for single-band images. In the case of multi-band images, a reasonable option is to

extract sources from the best quality band without considering other bands. However, that option ignores the multi-band information that can improve detection sensitivity.

- **Deblending Crowded Sources.** Existing methods eventually still rely on morphological forms of thresholding. This fact consequences under-segmented and over-segmented interacting sources at crowded regions, see Fig. 1.7 and Fig. 1.8. Especially, extended objects superimpose on top of larger objects are usually under-segmented because part of the extended object brightness is thresholded to the lower level components represent the large objects.

1.2.2 ConvNet-based Approaches and Limitations

Despite the early development stages, the ConvNet-based tools [Hausen and Robertson, 2020; Farias et al., 2020; Burke et al., 2019] have shown potential results comparing to the morphological-based tools. These tools are generally limited to applying computer vision models into astronomical datasets. The main approaches include U-net models for semantic segmentation and R-CNN models for instance segmentation.

Morpheus [Hausen and Robertson, 2020] - a pixel-level analysis framework was introduced to perform source detection, source segmentation, and morphological classification. Morpheus uses the U-net [Ronneberger et al., 2015] model to generate semantic segmentation for astronomical images. The fact is that semantic segmentation models classify each pixel to a particular label, but they are not trained to distinguish different instances of the same label. Hence, Morpheus later uses watershed to deblend the semantic output into separated objects. The U-net part was trained on fixed-size crops of real images from the CANDELS Survey [Koekemoer et al., 2011]. To handle large images, the framework performs a raster scan and accumulates semantic output crops. This simple strategy can approximate segmentation for such large images, but the accumulation usually produces non-realistic segmentation maps.

On the other hand, Astro R-CNN [Burke et al., 2019] addresses astronomical object detection with the Mask R-CNN [He et al., 2017] model on a simulation. Astro R-CNN completely relies on The PhoSim Simulator [Peterson et al., 2015] to generate both training and testing datasets. Even though the evaluation results of Astro R-CNN are very flattering on the simulation,

the simplicity and non-realisticity of the simulated images are questionable. In fact, Astro R-CNN performs poorly on real images and crowded regions which are different from the simulated images.

Mask Galaxy[Farias et al., 2020] is another tool toward the direction of using R-CNN models [Girshick et al., 2014]. Mask Galaxy narrows down the problem to detection, segmentation, and morphological classification of galaxies. Like Astro R-CNN, Mask Galaxy purely utilizes the well-known Mask R-CNN [He et al., 2017] but on a quantized real image dataset. The quantized dataset consists of JPEG images from Galaxy Zoo Catalog and Sloan Digital Sky Survey SDSS [Blanton et al., 2017]. Given the default Mask R-CNN model on the quantized dataset the loss of the JPEG compression, Mask Galaxy still shows reasonable detection results at galaxy detection, segmentation, and classification.

Limitations

In contrast, ConvNet-based methods [Hausen and Robertson, 2020; Burke et al., 2019; Farias et al., 2020] naturally adapt well with multi-band images (such as RGB). The two limitations of these approaches are *the availability of astronomical datasets* and *the adaptation to the astronomical context*.

- **Availability of Astronomical Dataset.** There is a lack of availability and consistency of standard astronomical datasets with labels for computer vision tasks. To date, the most well-known one is Galaxy Zoo with JPEG compressed images and crowded classification labels only. Some other small Catalogs (on CANDELS and SDSS images) provide sorts of variable-size crops with labels. As we can see, some tools tried to learn on non-realistic simulated datasets, but the models broke down when processing unseen real images.

- **Astronomical Context Adaptation.** Existing ConvNet-based source finders are at the early stage of applying computer vision models without considering the astronomical context. However, astronomical images are very different from natural images in terms of range, quantization technique, size, and other characteristics. Despite the robustness of applied U-net and applied R-CNN models on astronomical images with some positive results, the use of these models is still far from the practical needs of astronomers. We believe there is still room not just

to apply these models but to re-design the model architecture to adapt to the astronomical context.

1.2.3 State Of The Art

All in all, each tool has its pros and cons. While ConvNet-based source finders are still far from practical usages, a comparison [Haigh et al., 2020] has shown that MTObject achieves the highest scores on both area and detection measures among morphological-based tools. In this thesis, we use MTObject as the baseline for our experimental comparisons in Chapter 3 (Section 3.5) and Chapter 5 (Section 5.5).

1.3 Our Proposed Directions

To conclude, we state the primary interest of this thesis in Sec. 1.3.1. The following Sec. 1.3.2 explains the choices of our proposed directions to achieves these objectives.

1.3.1 Objectives

Our primary interest is to develop efficient object detection tools with application to the increasingly large astronomical datasets/surveys. We address three key requirements for our astronomical object detection models:

- **Automation and Easy to Use.** While the amount of astronomical data is increasing tremendously, it is critical to maximize automation. Many existing tools are reserved for experts with many hand-tuned parameters. We target to keep the tool simple, easy-to-use, and with few parameters.

- **Faint Structure Detection.** Faint structures are the most difficult part to find and to understand. Faint structures include faint objects and faint regions surrounding visible objects, see Fig. 1.6. These regions are the missing pieces of modern source finders, we target to detect them to improve detection performance.

- **Interacting Object Segmentation.** Existing source finders usually under-segment or over-segment objects located in interacting regions,

see Fig. 1.7 and Fig. 1.8. We target to achieve well-deblended objects in these crowded regions.

1.3.2 Directions

To design efficient astronomical object detection models, we relies on three main ideas: *Component-graphs*, *ConvNets*, and *Astronomical Context*. Now, we are going to relate each idea to the three objectives mentioned in Sec.1.3.1.

- **Component-graphs.** The information gain of the multi-band images is useful to improve both object detection and segmentation, see Fig. 1.5 and Fig. 1.6. For object detection, the information gain gives us more confidence in detecting faint structure that lies near the background level. For object segmentation, the color information of the multi-band images helps to deblend interacting regions.

  Despite the usefulness of the multi-band images, handling them is difficult and expensive. This is one of the reasons why most existing source finders (except ConvNet-based models) only process each sing-band image separately while the multi-band one is available.

  To take advantage of the multi-band images, we propose to use component-graph structures in our model, see Chapter 3. Compared to classical component-trees, such component-graphs [Passat and Naegel, 2014] are more general and more powerful at the cost of higher construction and filtering complexities.

- **ConvNet.** We have chosen to integrate ConvNet into our models to improve both object detection and segmentation, see Chapter 5. Before going to the advantages of ConvNet, we note that ConvNet-based models require labeled training datasets, which makes these ConvNet-based models less practical.

  On the other hand, the ConvNet architectures can naturally process multi-band images. ConvNet has shown excellent results in visual perception tasks.

  In contrast to morphology, ConvNet does not limit segmentation masks to the thresholded components, then we have some degree of freedom to define and optimize CNN-based models that allow overlapping segmentation. Furthermore, the output of the ConvNet models can be

used as soft segmentation masks where each pixel has a probability to belong to a mask.

- **Astronomical Context.** Astronomical images are very different from natural images in terms of range, quantization, size, and other characteristics. We see that many existing source finders just apply computer-vision models without considering these differences. These tools may work, but they can be improved. We target to tailor the base models with characteristics of the astronomical context.

  In astronomy, we observe that the center of the sources is usually brighter and better localized than the outer parts, i.e., the center is more important than the outer parts. We call it *centralization characteristic*.

  We have used that observation in several elements in our proposed models. In Chapter 3, the centralization characteristic is used to differentiate duplicated components in the component-graph. In Chapter 5, the same characteristic is used in CC-NMS module to detect multiple detections and being used in a smoothness regularizer. Also, the difference between astronomical and natural images motivates the development of a normalization layer in Chapter 5.

The methodological developments in this manuscript are in two main directions of Morphology and ConvNet, present as two parts:

- PART I. After providing the basis of the morphological connected operator in Chapter 2, our proposed morphological model, called CGO, is presented in Chapter 3. The CGO model uses component-graphs to bring astronomical object detection to the multi-band context.

- PART II. After an overview of ConvNet-based object detectors in Chapter 4, Chapter 5 turns our attention to the direction of ConvNet with two models: A ConvNet-based model and a hybrid model.

Bibliography

Akhlaghi, M. and Ichikawa, T. (2015). Noise-based detection and segmentation of nebulous objects. *The Astrophysical Journal Supplement Series*, 220(1):1.

Bertin, E. and Arnouts, S. (1996). Sextractor: Software for source extraction. *Astronomy and Astrophysics Supplement Series*, 117:393–404.

Blanton, M. R., Bershady, M. A., Abolfathi, B., Albareti, F. D., Prieto, C. A., Almeida, A., Alonso-García, J., Anders, F., Anderson, S. F., Andrews, B., et al. (2017). Sloan digital sky survey iv: Mapping the milky way, nearby galaxies, and the distant universe. *The Astronomical Journal*, 154(1):28.

Burke, C. J., Aleo, P. D., Chen, Y.-C., Liu, X., Peterson, J. R., Sembroski, G. H., and Lin, J. Y.-Y. (2019). Deblending and classifying astronomical sources with mask r-cnn deep learning. *Monthly Notices of the Royal Astronomical Society*, 490(3):3952–3965.

Doi, M., Tanaka, M., Fukugita, M., Gunn, J. E., Yasuda, N., Ivezić, Ž., Brinkmann, J., de Haars, E., Kleinman, S., Krzesinski, J., et al. (2010). Photometric response functions of the sloan digital sky survey imager. *The Astronomical Journal*, 139(4):1628.

Farias, H., Ortiz, D., Damke, G., Arancibia, M. J., and Solar, M. (2020). Mask galaxy: Morphological segmentation of galaxies. *Astronomy and Computing*, page 100420.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). 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.

Haigh, C., Chamba, N., Venhola, A., Peletier, R., Doorenbos, L., Watkins, M., and Wilkinson, M. (2020). Optimising and comparing source extraction tools using objective segmentation quality criteria. *Astronomy and Astrophysics*.

Hausen, R. and Robertson, B. E. (2020). Morpheus: A deep learning framework for the pixel-level analysis of astronomical image data. *The Astrophysical Journal Supplement Series*, 248(1):20.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

Koekemoer, A. M., Faber, S., Ferguson, H. C., Grogin, N. A., Kocevski, D. D., Koo, D. C., Lai, K., Lotz, J. M., Lucas, R. A., McGrath, E. J., et al. (2011). Candels: the cosmic assembly near-infrared deep extragalactic legacy survey—the hubble space telescope observations, imaging data products, and mosaics. *The Astrophysical Journal Supplement Series*, 197(2):36.

Kutner, M. L. (2003). *Astronomy: A physical perspective*. Cambridge University Press.

McLean, I. S. (2008). *Electronic imaging in astronomy: detectors and instrumentation*. Springer Science & Business Media.

Passat, N. and Naegel, B. (2014). Component-trees and multivalued images: Structural properties. *JMIV*, 49:37–50.

Peterson, J., Jernigan, J., Kahn, S., Rasmussen, A., Peng, E., Ahmad, Z., Bankert, J., Chang, C., Claver, C., Gilmore, D., et al. (2015). Simulation of astronomical images from optical survey telescopes using a comprehensive photon monte carlo approach. *The Astrophysical Journal Supplement Series*, 218(1):14.

Robotham, A., Davies, L., Driver, S., Koushan, S., Taranu, D., Casura, S., and Liske, J. (2018). Profound: source extraction and application to modern survey data. *Monthly Notices of the Royal Astronomical Society*, 476(3):3137–3159.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Teeninga, P., Moschini, U., Trager, S. C., and Wilkinson, M. H. (2016). Statistical attribute filtering to detect faint extended astronomical sources. *MMTA*, 1.

Wilkinson, M. H. F., Haigh, C., Gazagnes, S., Teeninga, P., Chamba, N., Nguyen, T. X., Talbot, H., Najman, L., Perret, B., Chierchia, G., Venhola, A., and Peletier, R. (2019). Sourcerer: A robust, multi-scale source extraction tool suitable for faint and diffuse objects. In *IAU 355 Symposium*.

Part I

MATHEMATICAL MORPHOLOGY FOR ASTRONOMICAL OBJECT DETECTION

Chapter 2

Morphological Connected Operators

This chapter presents an overview of *Connected Operators* in mathematical morphology that is the main context of our proposed morphological approach for astronomical object detection. Sec. 2.1 covers the historical development of connected operators from the early stage on binary images to the extensions on grey-scale and multi-band images. The extensions of connected operators motivate modernized hierarchical representations of images, including component-trees (Min-Tree, Max-Tree, Binary Partition Tree, Tree-of-Shapes) and component-graphs. Sec. 2.2 describes these hierarchical representations while Sec. 2.3 and Sec. 2.4 review primary construction algorithms and filtering strategies on these structures. Besides, this thesis explicitly interests in multi-band images, then we further discuss several advances of connected operators to handle multi-band images, including Component-graphs, Multivariate Tree-of-Shapes, and Connected Component-Tree in Sec. 2.6.

Contents

2.1 Introduction to Connected Operators

Mathematical morphology was first introduced by [Serra, 1982] for image analysis. Morphology can be used more generally in filtering, segmentation, classification, and analysis. It can handle various image types, from binary images to grey-scale and multi-band images.

In the field of mathematical morphology, *Connected Operator* is a class of morphological operators for digital image processing [Salembier and Wilkinson, 2009; Salembier and Serra, 1995]. Connected operators rely on the concept of connected components. The connected components are maximal sets of vertices in which a path exists between any two vertices. These operators do not act on individual pixels but merge and remove elements on connected

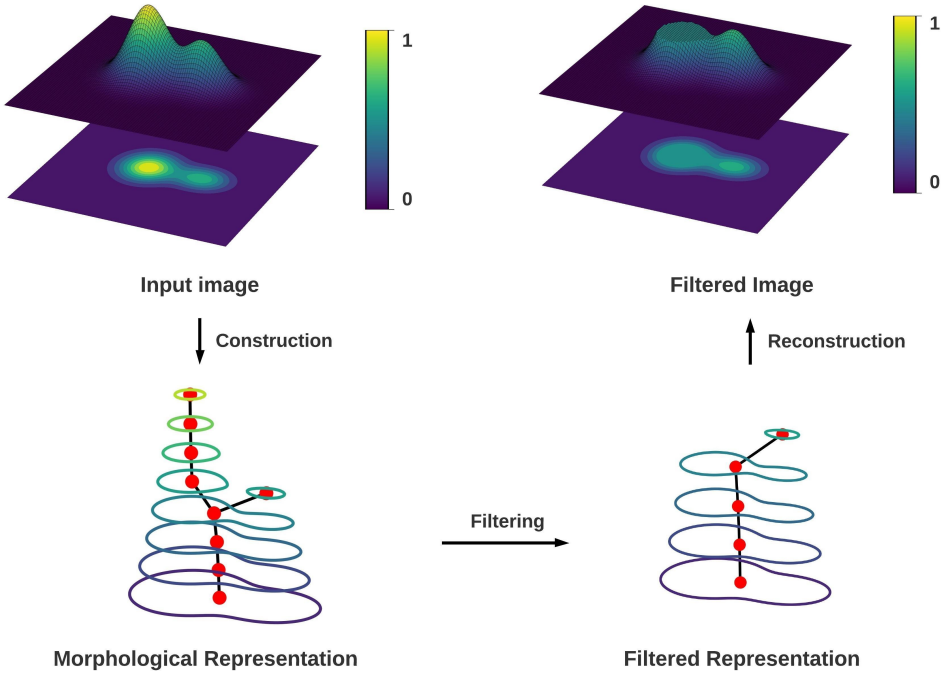
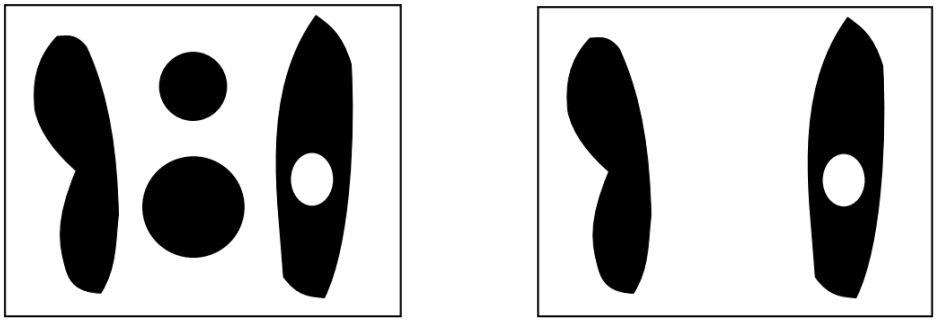


Figure 2.1: A schematic overview of connected operators.

component spaces. As a consequence, they do not create or move any contours, i.e., the connected operators are capable of preserving the contour information, which is essential in recognition and segmentation applications [Wilkinson and Westenberg, 2001; Berger and et al., 2007; Kurtz et al., 2012; Naegel and Wendling, 2010; Alonso-González et al., 2012].

A schematic overview of connected operators for grey-scale and multi-band images can be considered in three steps, as shown in Fig. 2.1:

- **Construction**: builds morphological representations (tree/graph) of the input image.
- **Filtering**: computes node attributes and selects relevant nodes in the hierarchical representation with attribute filtering strategies.
- **Reconstruction**: transforms selected nodes in to knowledge spaces, such as filtered image or segmentation.



(a) Original binary image. (b) Output binary image.

Figure 2.2: Example of binary connected operator [Najman and Talbot, 2013].

The first form of the connected operator is binary opening defined on binary images [Klein, 1976]. Given connected components of a binary image and a structuring component, the binary opening operator removes components by erosion with the given structuring component and leaves remaining components perfectly unchanged. The binary connected operator has been used for image simplification with contour preservation. An example of binary connected operator is illustrated in Fig. 2.2.

Later, the connected operator has been generalized for grey-scale images with opening by reconstruction [Vincent, 1993b], area opening [Vincent, 1993a], dynamics filters [Grimaud, 1992], and volumic filters [Salembier and Serra, 1995; Oliveras and Salembier, 1996]. An example of grey-scale connected operator is depicted in Fig. 2.3.

Recently, much more attention is extended to multi-band images with CC-Trees for multi-variate images [Perret et al., 2010], Multi-variate Tree-of-Shapes [Carlinet and Géraud, 2015], and Component-graphs [Naegel and Passat, 2014].

2.2 Morphological Representations of Images

The core idea of morphology relies on region-based processing where the image is viewed as a structured representation made of connected components. The connected component is defined on image graphs. In contrast to the classical grid of pixel representation, the image is now modeled as a graph

(a) Original image.

(b) Median filter.

(c) Area opening.

Figure 2.3: Example of grey-scale connected operator [Najman and Talbot, 2013].

with adjacency relations (the edges) between pixels (the vertices). The vertex values accompany pixels information, such as pixel intensity, while the edge weights encode pixel relationship measures, such as Euclidean distance. The connected components are maximal sets of vertices in these graphs in which a path exists between any two vertices. The concept of connected components plays the center role in building morphological representations of the image. Intuitively, Fig. 2.4 presents an example grey-scale image with several widely known morphological representations.

For the sake of completeness, we provide an overview of morphological structures in Sec. 2.2.1. The following sections recall preliminary definitions of order relations, connected components, and vertex-valued graphs to formally define the component-trees and the component-graphs. Visually, Fig. 2.5 and Fig. 2.6 show an example of a Max-Tree constructed from a single-band image and component-graph variants building on a two-band

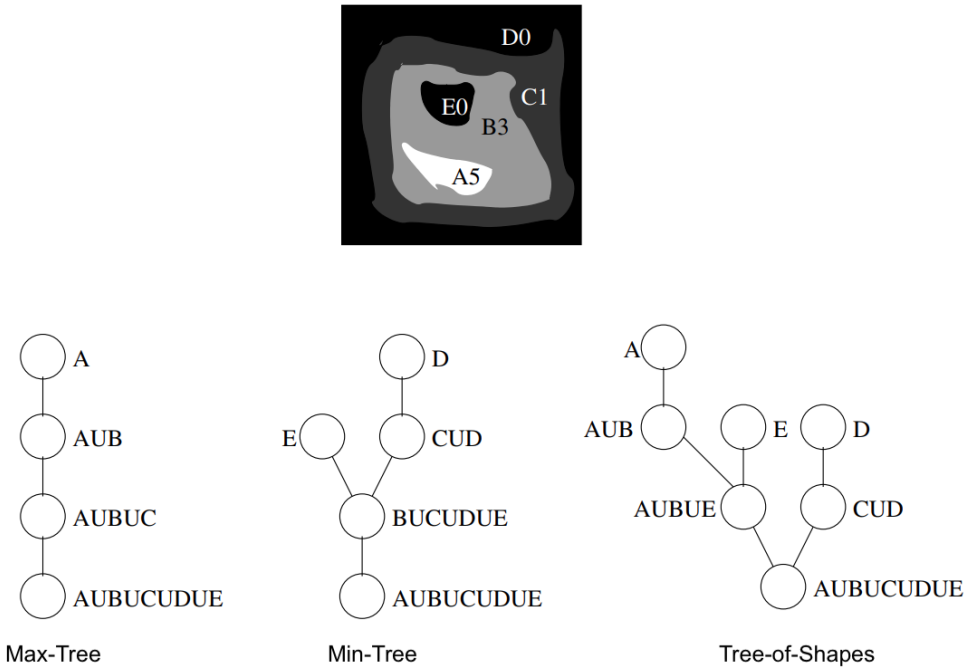


Figure 2.4: Tree representations of a grey-scale image [Najman and Talbot, 2013].

image.

2.2.1 Overview of Morphological Representations

Generally, morphological-based image representations include component-trees (Min-Tree, Max-Tree, Tree-of-Shapes), component-graphs, and multi-variate extensions of component-trees.

For grey-scale images, Min-Tree/Max-Tree [Salembier et al., 1998; Breen and Jones, 1996] are based on the inclusion relationship between peak connected components to represent the image. Instead of using the peak components, Tree-of-Shapes [Monasse and Guichard, 2000; Carlinet and Géraud, 2015] handles topological boundaries of the connected components by the upper/lower level sets, i.e., it merges the concept of Max-Tree and Min-Tree.

For multi-band images, Component-graphs [Passat et al., 2019] bring the ideas of the component-trees into multi-band context using partial orders

to model connected components as directed acyclic graphs. In contrast, other multivariate extensions of the component-trees, including Connected Component Trees CC-Trees [Perret et al., 2010; Perret and Collet, 2015] and Multivariate Tree-of-Shapes MToS [Carlinet and Géraud, 2015], introduce tree-based frameworks to handle multi-band images.

Since this thesis directly involves multi-band image processing, the following sections are dedicated to the multi-band extensions (Component-graphs, MToS, CC-Trees) and the transitions from grey-scale to multi-band representations. For grey-scale image representations, formal descriptions can be found in [Najman and Talbot, 2013; Salembier and Wilkinson, 2009].

2.2.2 Order Relations

Order relations are essential to define the relationships between components in the morphological structures. Given a finite set of elements Γ, a binary relation $\leq$ on Γ is an order relation and $(\Gamma, \leq)$ is a finite ordered set if $\leq$ is reflexive, transitive, and anti-symmetric.
Formally, $\forall x, y, z \in \Gamma$:

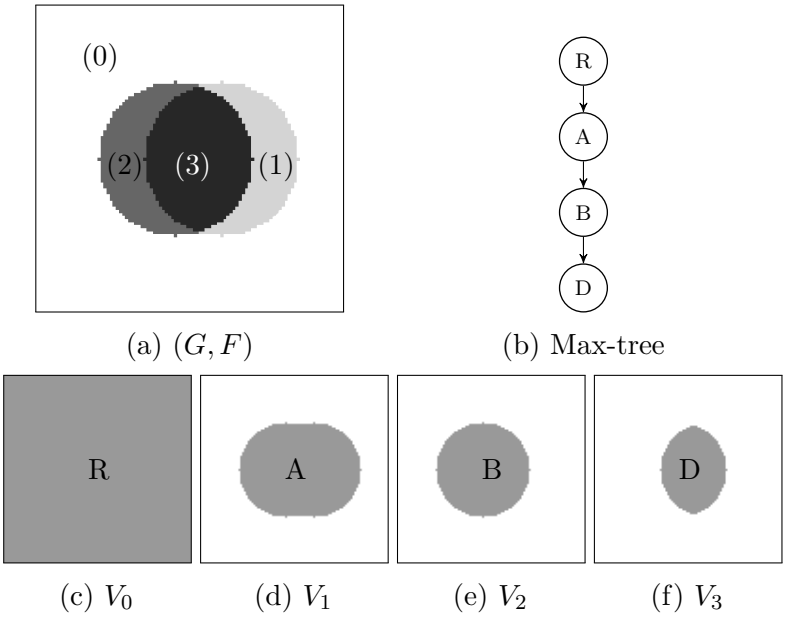
$$x \leq x;$$
$$(x \leq y \wedge y \leq z) \Rightarrow (x \leq z);$$
$$(x \leq y \wedge y \leq x) \Rightarrow (x = y).$$

We say that $\leq$ is a partial order relation, and that $(\Gamma, \leq)$ is a partially ordered set, if there exist non-comparable elements in $(\Gamma, \leq)$, i.e., $\exists x, y \in \Gamma, (x \not\leq y \wedge y \not\leq x)$. The order relation $\leq$ is a total order relation, and $(\Gamma, \leq)$ is a totally ordered set, if $\forall x, y \in \Gamma, (x \leq y \vee y \leq x)$. A finite ordered set can be uniquely represented by Hasse diagram, which is the transitive reduction of the ordered set.

2.2.3 Connected Component and Vertex-valued Graph

A *graph* G is a pair (V, E), where V is a finite set and E is a set of pairs of distinct elements of V, i.e., $E \subseteq \{\{x, y\} \subseteq V \mid x \neq y\}$. An element of V is called a *vertex* of G, an element of E is called an *edge* of G.

Given a graph $G = (V, E)$ we say that a sequence of elements $(x_0, \ldots, x_n) \in V$ is a *path* in V from x_0 to x_n if $\{x_{i-1}, x_i\} \in E, \forall i \in \{1, \ldots, n\}$. A subset $V' \subseteq V$ is said to be *connected* if for any two distinct elements $x, y \in V'$, there exists a path from x to y. A *connected component* of G is a maximal



(a) (G, F) (b) Max-tree

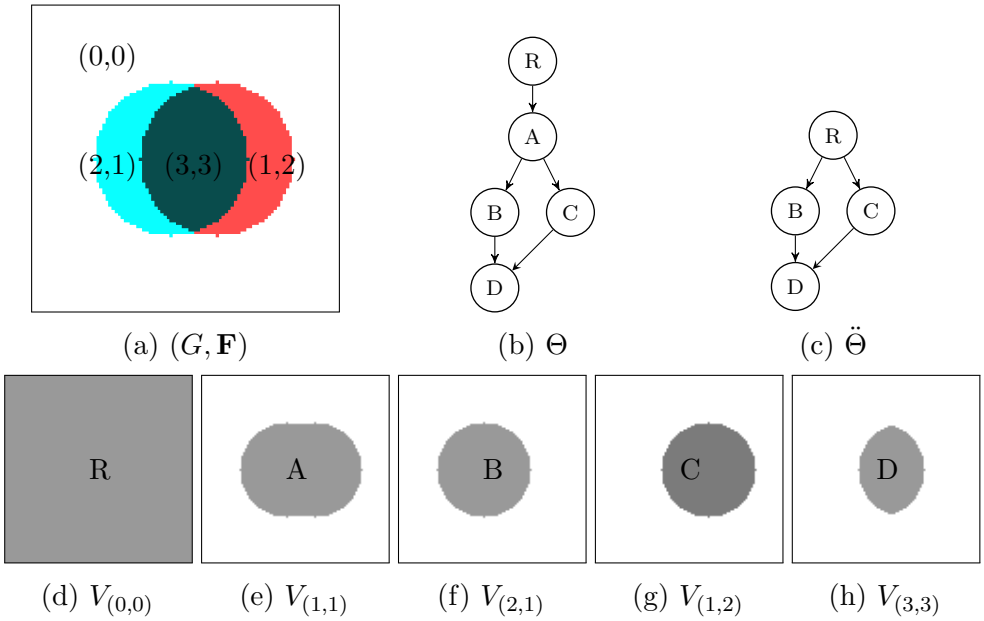
(c) V_0 (d) V_1 (e) V_2 (f) V_3

Figure 2.5: Component-tree example: (a) A grayscale image with values in $\mathbb{V} = \{0, 1, 2, 3\}$; (b) The Max-tree of the image; and (c-f) The threshold sets V_v for $v \in \mathbb{V}$. The letters (R, A, B, D) refer to the connected components corresponding to the nodes in the tree. Note that the connected components in figures (c-f) are down-scaled by a factor of two for visualization purpose.

connected subset of V. The set of all connected components of G, denoted as $C[G]$, is a partition of V.

Let $\mathbf{F}$ be a function from V to a nonempty set $\mathbb{V}$ equipped with an order relation $\leq$. We say that $(G, \mathbf{F})$ is a vertex-valued graph (or valued graph).

In practice, given a valued graph $(G, \mathbf{F})$, the graph G can be used to represent the domain of an image where each vertex corresponds to a pixel and where edges correspond to the adjacency relation between pixels [Kong and Rosenfeld, 1989]. The function $\mathbf{F}$ then represents an image associating a possibly multivariate value to any pixel/vertex. The order relation is now defined on intensity space.



(a) $(G, \mathbf{F})$ (b) Θ (c) $\ddot{\Theta}$

(d) $V_{(0,0)}$ (e) $V_{(1,1)}$ (f) $V_{(2,1)}$ (g) $V_{(1,2)}$ (h) $V_{(3,3)}$

Figure 2.6: Component-graphs example: (a) A two-band image with multivariate values in $\mathbb{V} = \{(0,0),(2,1),(1,2),(3,3)\}$ equipped with the marginal partial order relation $\leq_{\mathrm{m}}$; (b-c) The CG Θ and the simplified CG $\ddot{\Theta}$ of the image; and (d-h) The threshold sets V_v for $\boldsymbol{v} \in \mathbb{V}$. The CG $\ddot{\Theta}$ does not contain the node A because A is invisible (behind B and C) in the input image.

2.2.4 Component-Trees to Component-Graphs

Given a valued graph $(G, \mathbf{F})$, we define the threshold set

$$V_v = \{x \in V \mid \mathbf{F}(x) \geq \boldsymbol{v}\}, \tag{2.1}$$

where $\boldsymbol{v} \in \mathbb{V}$. The threshold set V_v induces a subset $E_v = \{\{x, y\} \in E \mid x, y \in V_v\}$ and a sub-graph $G_v = (V_v, E_v)$. The set of connected components of the sub-graphs G_v of G for all $\boldsymbol{v} \in \mathbb{V}$ is denoted as

$$\Psi = \bigcup_{\boldsymbol{v} \in \mathbb{V}} C[G_v]. \tag{2.2}$$

- If $(\mathbb{V}, \leq)$ is totally ordered, the partially ordered set $(\Psi, \subseteq)$ forms a *Max-Tree* of the valued graph $(G, \mathbf{F})$ (see Fig. 2.5).

- If $(\mathbb{V}, \leq)$ is partially ordered, the partially ordered set $(\Psi, \subseteq)$ forms a *component-graph*, denoted by Θ, of the valued graph $(G, \mathbf{F})$ [Passat and Naegel, 2014] (see Fig. 2.6).

2.2.5 Simplified Component-graph

[Naegel et al., 2007] has introduced a simplified version of the CG, denoted $\ddot{\Theta}$ (see Fig. 2.6c), where its set of connected components

$$\ddot{\Psi} = \left\{ X \in \Psi \middle| \bigcup_{\substack{Y \in \Psi \\ Y \subsetneq X}} Y \neq X \right\} \tag{2.3}$$

contains only the connected components that contribute to the visibility of the image $\mathbf{F}$ [Naegel and Passat, 2014]. The CG Θ and the CG $\ddot{\Theta}$ are both directed acyclic graphs. The set $\ddot{\Psi}$ is a subset of the set Ψ. The CG Θ associated to the set Ψ containing all valued connected components in the image is the most informative structure but also the most expensive to construct ($\mathcal{O}(n^3)$). Since the CG $\ddot{\Theta}$ takes into account only visible components from the image, it is less expensive to construct ($\mathcal{O}(n^2)$) than the full CG Θ [Grossiord et al., 2019]. Note that all three of CT, Θ and $\ddot{\Theta}$ are lossless representations of the same image, and so no information is lost despite the simplification.

2.2.6 Directed Connected Operators

Directed Connected Operators were proposed by [Perret et al., 2014], the framework generalizes the concept of *Connected Operators* from un-directed graphs to directed graphs. While connected operators rely on the symmetric definition of adjacency [Rosenfeld, 1970] to represent images as undirected graphs, directed connected operators consider non-symmetric adjacency relations to view images as directed-graphs. As a consequence, the induced morphological representations are no longer trees but directed acyclic graphs, as shown in Fig. 2.7.

Compared to the undirected graph, the directed graph is helpful in the way that it can naturally handle the directed information, such as weak connections or prior knowledge. Along with the advantage, the new framework led us to the new notion of directed connected component (D-component in

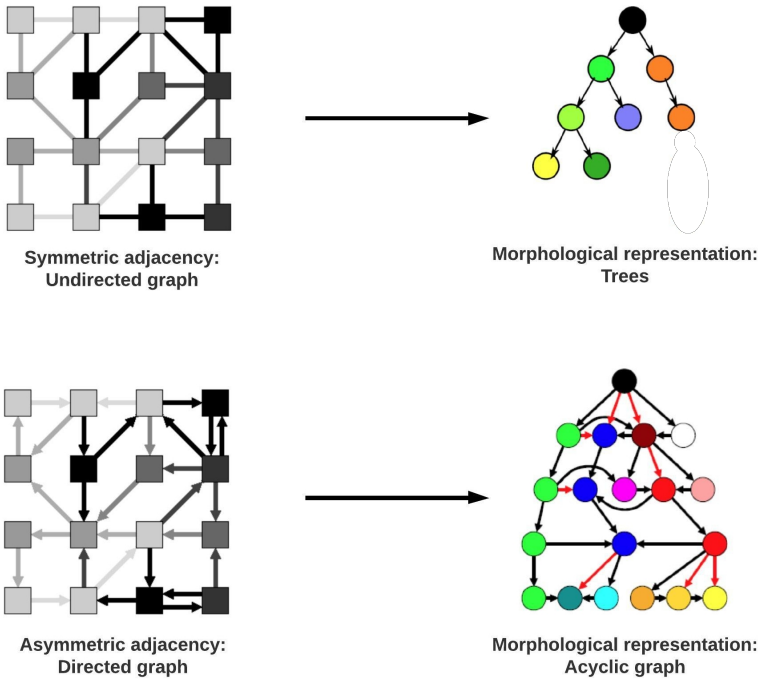


Figure 2.7: Directed Connected Operators versus Connected Operators [Perret et al., 2014].

short), which generalize the connected component (Sec. 2.2.3) to directed graphs. The formal definitions of the directed graph and the D-component are presented below.

Directed Graph Formally, a *directed graph* G_D is a pair of (V, A), where V is a nonempty finite set, and A is is composed of pairs of elements of V, i.e., A is a subset of $V \times V$. Each element of V is called a vertex or a node of G_D, while each element of A is called an arc of the direct graph G_D. Arc in the directed graph is equivalent to the edge in the undirected graph with direction information.

Let G_D be a directed graph, a *path* in G_D from a vertex $x \in G_D$ to a vertex $y \in G_D$ is a sequence $x_0, \ldots, x_l$ of vertices of G_D such that $x_0 = x, x_l = y$, and for any $i \in \{1, \ldots, l\}$, the pair (x_{i-1}, x_i) is an arc of G_D. We say that y is a successor of x in G_D, i.e., x is a predecessor of y in G_D.

D-component : Let G_D be a directed graph and x be a vertex of G_D. The *D-component* of basepoint x, denoted as $DCC_{GD}(x)$, is the set of all the successors of x in the directed graph G_D.

In contrast to the connected components, a vertex in the graph may belong to several D-components. In other words, the set of all D-components of a directed graph is not necessarily a partition of its vertex set. In contrast, the set of all connected components of an undirected graph is a partition of its vertex set.

The directed connected operator framework has shown useful for various filtering and segmentation tasks, such as neurite filtering and retina segmentation [Perret et al., 2014].

2.3 Construction Algorithms

This section recaps an overview of construction algorithms for the component-trees and the component-graphs.

2.3.1 Building Max-Trees

The component-trees (Min-Tree, Max-Tree [Salembier et al., 1998,?; Breen and Jones, 1996], Tree of Shape [Monasse and Guichard, 2000]) benefit from efficient construction and filtering algorithms [Carlinet and Géraud, 2014; Najman and Couprie, 2006; Géraud et al., 2013]. Various algorithms have been proposed, they can be grouped in three main approaches: Union-find-based, flooding-based algorithms, and merge-based algorithms.

Union-find-based algorithms

[Najman and Couprie, 2006; Géraud et al., 2013; Berger and et al., 2007] build the component-trees in bottom-up style, from leaves to root. The algorithm starts with a disjoint set of leaf components for each pixel. Afterward, it merges disjoint components to create upper-level components until reaching the root. The merging process can be done with the union-find algorithms [Tarjan, 1975].

Two important optimizations of the union-find algorithm are *path compression* and *union-by-rank*. When merging components, path compression

attempts to reduce the cost of querying the root of components while union-by-rank tries to avoid degenerated tree construction. In detail, path compression collapses the morphological representations of input image as intermediate mappings from leaf components to their current roots. This helps to query the current root of a component in constant time instead of iteratively backtracking in the hierarchy. On the other hand, union-by-rank helps to balance the tree when merging two components. It measures the rank of a component by the depth of the tree rooted in that component. The greatest rank component is selected to be the representative component of the union of two components. Besides, it is necessary to sort components by values during the construction process. Sorting can be done in $\mathcal{O}(n)$ if the component values are small integers (counting sort [Cormen et al., 2009]), and in $\mathcal{O}(nlog(n))$ for general cases. Union-by-rank and path compression in combination with sorting component algorithms can construct the component-trees in quasi-linear time $\mathcal{O}(n\alpha(n))$ where n denotes the summation of the number of vertices and the number of edges; and α is a very slow-growing diagonal inverse of the Ackermann's function (we have $\alpha(10^{80}) \approx 4$).

Flooding-based algorithms

In contrast, [Salembier et al., 1998; Wilkinson, 2011] design the flooding algorithms in a top-down manner. First, they scan over the whole input image to identify the root pixel as the root component. When scanning, sorted pixels are stored in a queue simultaneously. Then, a depth-first propagation from the root component to neighboring pixels is processed to form the lower branch of the tree. One can note that the neighboring pixels are addressed via adjacency connectivity, such as 4-connectivity or 8-connectivity.

For these flooding-based algorithms, a *priority queue* can reduce the general complexity. Instead of the regular queue, the priority queue saves the propagated pixels with pixel levels as priority values. During the flooding, the algorithm can quickly access the pixels at the highest priority value in the priority queue in constant time. In terms of complexity, given n elements, regular queues cost $\mathcal{O}(1)$ for insertion and $\mathcal{O}(n)$ for querying the min/max element while priority queues cost $\mathcal{O}(log(n))$ for insertion and $\mathcal{O}(1)$ to find the min/max. In the case of small integers, hierarchical priority queues can achieve $\mathcal{O}(1)$ for the both operations. Typically, priority queues are implemented with a tree-based heap structure.

Merge-based algorithms

They target to parallel the tree/graph construction for large input images [Wilkinson et al., 2008; Ouzounis and Wilkinson, 2007]. First, the input is partitioned into multiple slices. They then build the Max-Trees of each slice using any sequential Max-Tree algorithm. Most importantly, the Max-trees of the slices can be merged to obtain the Max-tree of the whole input image.

2.3.2 Building Component-graphs

The component-graph can be thought of as the extension of the component-tree to multi-band images. Building a component-graph shares similar ideas with building a component-tree, but there remain algorithmic difficulties because of the structural differences. Particularly, the component-graph is no longer a tree but a DAG [Naegel and Passat, 2014]. Because of the graph-structures, the key optimization techniques to construct component-trees (such as path compression, union-by-rank, and priority queue) are no longer applicable in the context of component-graph construction.

Sequential construction algorithms for the component-graph variants have been proposed by [Naegel and Passat, 2014; Passat et al., 2019]. For the sake of completeness, we recap the construction algorithms for the simplified variant $\ddot{\Theta}$ in Alg. 1, proposed by [Naegel and Passat, 2014].

In Alg. 1, the input includes an image viewed as a vertex-valued graph $G = (V, E)$ with a value function $\mathbf{F}$ and a priority function $priority()$. The idea is to build the component-graph in bottom-up style using the function $priority()$ to decide pixel/region visit ordering. The procedure has three main steps:

First, in *line(1-4)*, a region-adjacency graph (RAG) *rag* is computed from G. RAG is the graph representation of the input image, in which RAG vertices are pixels or flat-pixels and RAG edges represent vertices neighboring relationships. Flat-pixel is a group of neighboring pixels sharing the same values. We can think of RAG as of graph of flat-pixels compared to the usual graph of pixels. So, instead of building the component-graph from pixel level, we now start from flat-pixel level. The RAG sometimes can reduce the construction complexity.

Second, in *line 5-24*, based on the RAG rag, we visit the flat-pixels from leave to root with the help of the priority function. For each flat-pixel (*line 6*), a new node is created if the flat-pixel does not belong to any node (*line*

7-10). Now, we need to determine the relationship between the new node and existing nodes. The algorithm performs a neighboring propagation from the newly created node to: (a) find other flat-pixels belonging to the same node (*line 15-16*), and (b) find all the descendants of the new node (*line 18-23*).

Finally, all nodes and node relationships are reorganized in sub-graphs (*line 25-27*). A virtual node represents the whole image is added into the graph to connect all sub-graphs (*line 28*).

2.4 Filtering Strategies

Once tree-based and graph-based representations of images are constructed, connected filtering (or attribute filtering) plays the part of removing specific connected components while leaving relevant connected components. The filtering strategies act on connected components instead of individual pixels. Roughly speaking, such filtering is equivalent to an attribute function $\mathcal{STR}$ on attribute space $\mathcal{ATT}$ (criterion) of the connected components of morphological representation $\mathcal{T}$. For instance, if we want to remove elongated regions from the Max-Tree of a single-band image, we can first define any elongation attribute that numerically states how elongated a component is; then we can define an attribute function to simply eliminate less elongated components from the Max-Tree; we can then reconstruct the filtered image from the remaining components of the Max-Tree.

The design of the attribute function and the attribute space are flexible and depend on the characteristic of the input image and task-specific applications. Generally, component attributes $\mathcal{ATT}$ are categorized as increasing and non-increasing attributes.

2.4.1 Increasing Attribute Filtering

Increasing Attributes, denoted as $\mathcal{ATT}^{\uparrow}$, meaning that if a connected component A is descendent of connected component B, then the attribute of A is smaller than the attribute of B. Formally, for a node N in the representation $\mathcal{T}$, we have

$$\forall \mathrm{N} \in \mathcal{T}, \mathcal{ATT}^{\uparrow}(N) \leq \mathcal{ATT}^{\uparrow}(\mathrm{parents(N)}), \tag{2.4}$$

where parents(N) is the set of parents of the node N.

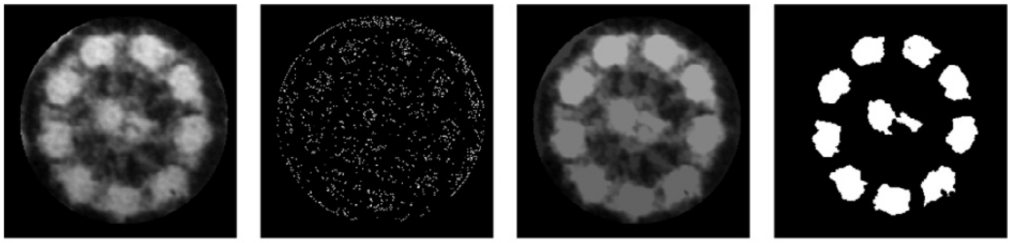


Figure 2.8: Increasing attribute filtering example [Najman and Cousty, 2014]: (a) Original image; (b) Maxima of image (a), in white; (c) Image filtered with an increasing criterion (volume) on the Max-Tree. (d) Maxima of image (c), which correspond to the ten most significant lobes of the image (a).

Some examples of increasing attributes are component area, component level in the Min/Max-Tree, component volume (the sum of the belonging pixel values), or diameter of the largest/smallest circle that can fit/enclose a component.

It is simple and straightforward to design thresholding filtering strategies on the increasing attributes. In detail, given an increasing attribute, if a node attribute does not satisfy the filtering criterion, then all of its descendants are guaranteed to fall into the same situation. In that case, the filtering simply cuts the branch (the node and its descendants).

An example of increasing attribute filtering is presented in Fig. 2.8 where the node volume attribute is used to filter small nodes in the Max-Tree of the input image.

2.4.2 Non-Increasing Attribute Filtering

Non-increasing Attributes mean that if a connected component is a descendent of another connected component, then there is no constraint between the attribute values of the two components. In practice, the majority of useful component attributes are not increasing. To name a few, compactness, elongation, roundness, sharpness, and perimeter components are all non-increasing attributes.

An example of node circularity - a non-increasing attribute is illustrated in Fig. 2.9 (a-f). As we can see, the two visually similar circles in the input

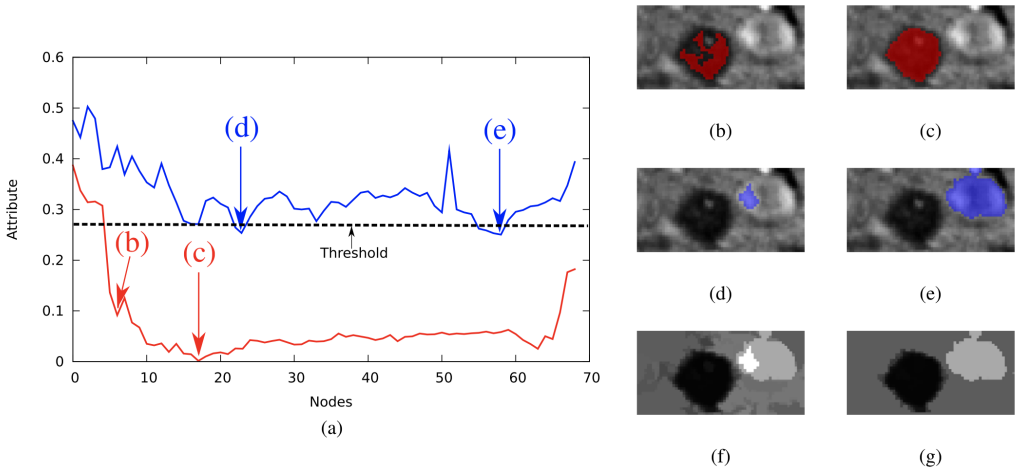


Figure 2.9: Non-Increasing attribute filtering example [Najman and Cousty, 2014; Xu et al., 2015]: (a) Evolution of a 'circularity' criterion on two branches of a tree of shapes; (b–e): Some shapes; (f) Attribute thresholding; (g) A morphological shaping, detail in Section 2.4.3.

image appear differently via the circularity attribute.

In these non-increasing circumstances, classical thresholding is not robust, several strategies have been proposed by [Salembier, 2013; Salembier et al., 1998; Salembier and Wilkinson, 2009; Urbach et al., 2007] to cut the whole branches of the tree with the rules taking into account both current component attribute value and its ancestor or descendant components (Min, Max, Viterbi, Merging). Given a node N in the hierarchy, its ancestors N_{anc}, its descendants N_{des}, and a filtering criterion threshold t, the filtering strategies are defined as follows:

- Min: The node N is removed in two cases: either $\mathcal{ATT}(\mathrm{N}) < t$ or there exists an ancestor of N such that $\mathcal{ATT}(\mathrm{N}) < t$.
- Max: The node N is removed if $\mathcal{ATT}(\mathrm{N}) < t$ and all its descendants satisfy $\mathcal{ATT}(\mathrm{N}_{des}) < t$.
- Viterbi: This strategy models the node selection process as a cost optimization problem with Viterbi algorithm [Viterbi, 1967]. Assigning the cost of nodes is based on node removal and node preservation decisions. From leaf to root, each transition of removal decision is assigned a cost.

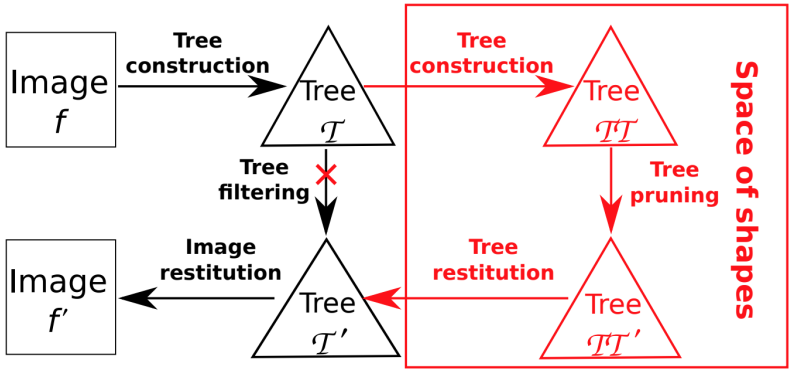


Figure 2.10: Shaping framework: Classical connected operators (black path) and connected operators on tree-based shape spaces (black+red path) [Xu et al., 2015]

Each transition of preservation decision to the removal of its parents is penalized heavily with infinite cost. In other words, this strategy does not allow to preserve a node while removing its ancestors [Salembier et al., 1998]. Then, each leaf node is associated with the branch which has the lowest cost.

- Direct and Subtractive: *Direct* approach flattens the removed nodes by the lowest preserved ancestor [Salembier et al., 1998] while *Subtractive* subtracts node value difference by all its descendants to preserve the local contrast [Breen and Jones, 1996]. More details in Section 2.5.

2.4.3 Filtering with Shaping

Shaping [Xu et al., 2012, 2015] is another approach to deal with non-increasing attributes. An example of Shaping dealing with non-increasing circularity attribute is depicted in Fig. 2.9 (g). The idea of shaping is to build a second tree $\mathcal{TT}$ on the first morphological representation $\mathcal{T}$ of the input image. The overview of the shaping framework is presented in Fig. 2.10, given an input image f, the process consists of five steps:

- First Tree Construction: There are two trees in the shaping framework. The first tree $\mathcal{T}$ of the input image is constructed the same as the

classical filtering framework. The tree $\mathcal{T}$ can be component-trees (such as Min-Tree, Max-Tree, Tree-of-Shapes) or component-graphs.

- Second Tree Construction: Shaping considers $\mathcal{T}$ as a valued-graph whose edges are defined by the parenthood relationship, and vertex weights are the non-increasing attribute themselves. The second tree $\mathcal{TT}$ is a component-tree built on the value-graph. The second tree is called shape space.

- Shape Space Filtering: Node attributes in the second tree $\mathcal{TT}$ are designed to be increasing, then filtering $\mathcal{TT}$ is straightforward as described in Sec. 2.4.1. Selected nodes are saved in $\mathcal{TT}'$.

- Tree Reconstruction: Shaping requires first reconstructing the simplified tree $\mathcal{T}'$ from $\mathcal{TT}'$. To obtain $\mathcal{T}'$, it is trivial to remove nodes on $\mathcal{T}$ corresponding to the removed nodes on $\mathcal{TT}'$.

- Image Reconstruction: Similar to classical filtering, an image can be reconstructed from the simplified tree $\mathcal{T}'$ using reconstruction rules described in Sec. 2.5.

The shaping framework has been applied to several applications of image simplification [Xu et al., 2012] and segmentation [Grossiord et al., 2015]. Later, the idea of shaping has been extended to building a tree on a graph with application to PET image segmentation [Grossiord et al., 2019].

2.5 Reconstruction

The reconstruction process is primarily about assigning values for the pixels belongings to the removed nodes. The process varies a lot depending on the previous filtering stage.

In the case of increasing attribute filtering, eliminating a node will cut all of its descendants while leaving its ancestors intact. Since the descendant branches are completely removed, then the reconstruction process is straightforward; it assigns the value of the lowest preserved node to pixels belonging to the removed nodes.

In the case of non-increasing attribute filtering, the selected nodes can be located anywhere in the representation $\mathcal{T}$, i.e., the descendants of the the removed nodes may be preserved. Based on the choices of the removed

node values, there are two main reconstruction rules: *Direct* [Salembier et al., 1998] and *Subtractive* [Breen and Jones, 1996; Urbach et al., 2007; Wilkinson and Westenberg, 2001].

2.5.1 Direct Reconstruction

This rule flattens the pixels belonging to the removed nodes by the value of the lowest preserved ancestor. This approach is similar to the case of increasing attribute filtering reconstruction. Note that the direct approach does not maintain the local contrast of the preserved nodes. Parts of these local contrast are revealed on the residual image, which is the subtraction of the reconstructed image by the original image.

2.5.2 Subtractive Reconstruction

This rule subtracts pixel values by the value of the lowest preserved ancestor. Consequently, the local contrast of pixels belonging to the removed nodes is presented in the reconstructed image. In this case, the residual image will present the same removed nodes in the representation $\mathcal{T}$.

2.6 Multi-band Image Processing Perspectives

This section covers multi-band image processing in mathematical morphology. So far, the component-trees have shown that they can handle grey-scale images efficiently while the component-graphs can fully support multi-band images, see Sec. 2.2. We further briefly review the extension of mathematical morphology to multi-band data in Sec. 2.6.1. Then, we discuss recent advances in the field to tackle multi-band images, including Connected Component-Tree [Perret et al., 2010] in Sec. 2.6.2, Multivariate Tree-of-Shapes [Carlinet and Géraud, 2015] in Sec. 2.6.3, and Component-Graphs in Sec. 2.6.4.

2.6.1 Overview of MM to multi-band Data

In the mathematical morphology framework, two main approaches are usually proposed to perform multi-band image processing: *marginal processing* and *vectorial processing*, as shown in Fig. 2.11.

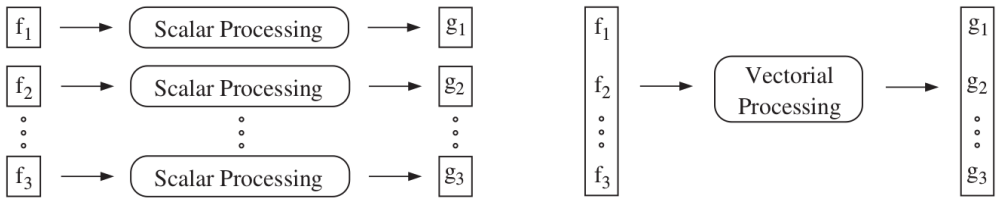


Figure 2.11: (left) Marginal processing and (right) Vectorial processing strategies [Aptoula and Lefèvre, 2007].

- *Marginal Processing*: This approach processes each channel separately, thus reducing the problem to the processing of grey-scale images. This strategy can leverage all algorithms offered by grey-scale morphology. However, the repetition of the marginal processing on each band make it computational expensive. Beside, the inter-band correlation (i.e., the multi-band information gain) is ignored [Aptoula and Lefèvre, 2007].

- *Vectorial Processing*: This approach processes all available bands globally and simultaneously. It requires defining a total order (or pre-order) relation on the set of multi-band components. Several vector-based orderings were proposed, including conditional ordering (C-ordering, including lexicographic ordering), reduced ordering (R-ordering, which implies to reduce a vector value to a scalar one), and partial ordering (P-ordering, where vectors are gathered into equivalence classes) [Naegel and Passat, 2009].

  Compared to the marginal counterpart, the drawback of the vectorial approach is the need for adapting the existing algorithms to accommodate vectorial data. Thus leading often to slower implementations than their marginal processing.

2.6.2 Connected Component Tree (CC-Trees)

[Perret et al., 2010; Perret and Collet, 2015] have proposed a general framework to process multi-band images with the component-trees, call the *CC-Trees* framework. To be clear, the CC-Trees framework does not introduce any new tree structure, it generalizes and discusses the use of component-trees in the multi-band contexts.

In the CC-Trees framework, the most important thing is to define an order relation on the multi-band space. The choice of order relation influences how to build and reconstruct the tree representations of the multi-band input images. In the case of total order, the process is similar to handling grey-scale images. On the other hand, if the order is a total pre-order, then the anti-symmetry relation between connected component levels is not guaranteed. For the latter case, the reconstruction process could not use the component levels directly because multiple values may associate with a single component. [Naegel and Passat, 2009] proposed to use a new representative value (such as the mean or the median) of the multiple levels of a component. However, the new level may introduce new values or require another total order.

The CC-Trees framework has been applied to detect multi-band astronomical sources. To process multi-band astronomical images, [Perret et al., 2010] defined a vectorial order relying on a reduced-order and a lexicographic order. Formally, let $v, v' \in \mathbb{R}^n$, the pre-order $\leq_{A_p}$ is defined as

$$\begin{aligned} & v \leq_{A_p} v' \\ \Leftrightarrow & \left[\lfloor E_n(v) \rfloor, \left\lfloor \frac{v_1}{k\sigma_1} \right\rfloor, \ldots, \left\lfloor \frac{v_n}{k\sigma_n} \right\rfloor\right] \leq_L \left[\lfloor E_n(v') \rfloor, \left\lfloor \frac{v'_1}{k\sigma_1} \right\rfloor, \ldots, \left\lfloor \frac{v'_n}{k\sigma_n} \right\rfloor\right], \end{aligned} \tag{2.5}$$

where n is number of band, $\leq_L$ is the lexicographic order, $\sigma_1, \ldots, \sigma_n$ are standard deviation of the noise in each band, k is a confidence factor, and normalized energy $E_n(v)$ term is defined as

$$E_n(v) = \left\| \lfloor E_n(v') \rfloor, \left\lfloor \frac{v'_1}{k\sigma_1} \right\rfloor, \ldots, \left\lfloor \frac{v'_n}{k\sigma_n} \right\rfloor \right\|. \tag{2.6}$$

2.6.3 Multivariate Tree of Shapes (MToS)

[Carlinet and Géraud, 2015] proposed to build MToS - a tree-based representation of multi-band images. The MToS does not impose any arbitrary ordering on values, but it is purely based on the inclusion relationship between connected components in the Tree-of-Shapes (ToS). MToS's main idea is combining and deducting multiple ToS's computed marginally on each image band, as shown in Fig. 2.12. Given the multi-band input, building MToS consists of two main phases:

- *Graph Computation*: The first phase combines marginal ToS's built on each separated image band as a single graph of shape G (GoS).

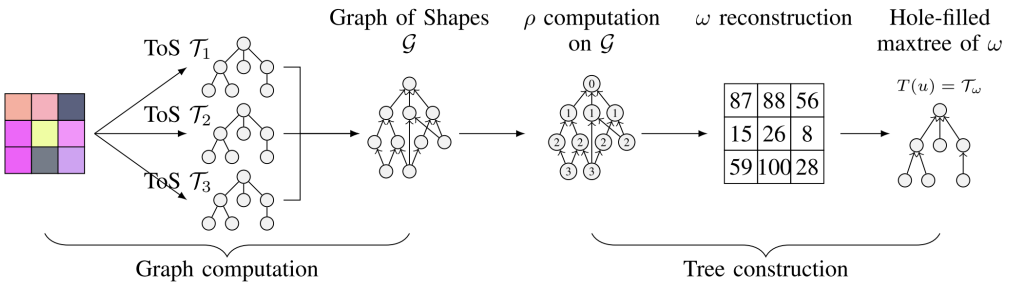


Figure 2.12: MTOS Overview in 5-steps [Carlinet and Géraud, 2015]: (1) Tree-of-Shapes (ToS) are computed marginally on each band of the input image; (2) The ToS's are merged into the GoS G; (3) an algebraic attribute is computed on G; and(4) yields a scalar attribute map ω, (5) a final tree is built upon ω.

The graph of shape G contains the set of marginal components of the TOS's and the inclusion relation between the components. Formally, let $S_1, \ldots, S_n$ be sets of components of the marginal ToS $T_1, \ldots, T_n$ and $S = \bigcup_1^n (S_i)$, the graph of shape G is the cover of $(S, \subseteq)$.

- *Tree Construction*: The second phase deducts a tree presentation from the graph of shape G. The key is to get a new set of components from G that do not overlap, and then we can extract a valid morphological tree from them. [Carlinet and Géraud, 2015] has proved that the Max-Tree of the depth map ω yields the Tree-of-Shapes of the non-overlapping components, called MToS. The depth map ω associates each pixel with the depth of the deepest component containing it. Formally,

$$\omega(x) = \max_{X \in S, x \in X} \rho(X), \tag{2.7}$$

where ρ is the depth attribute of the component. However, the depth map may form components with holes, a hole-filling step is necessary to guarantees valid components in the final tree.

The MToS has involved different kinds of multivariate data, ranging from color images and videos, hyperspectral data to multimodal medical images. Despite the simplicity of having a single tree-based representation of multi-band data, MToS faces reconstruction problems like CC-Trees. Particularly, the graph of shape GoS is a complete representation of the input image while

MToS is not. The combining phase has to accept some loss to induce a single tree-representation from the GoS.

2.6.4 Component-graphs

This section discusses the potential of the component-graphs at handling multi-band images. We have chosen the component-graphs [Passat et al., 2019; Grossiord et al., 2019; Naegel and Passat, 2014] as a base structure to build one of our proposed approach in Chapter 3. For formal definition of the component-graphs, see Sec. 2.2.4.

Beyond the classical multivariate extensions of the component-trees, the component-graph efficiently holds the whole structural information of multi-band images as directed acyclic graphs (DAGs). Such DAGs are more general and more powerful at the cost of higher construction and filtering complexities. The richness of the component-graphs are expected to lead to increased sensitivity, i.e., the information gain can be used to go deeper into the noise to find objects near the background levels. The challenge is to effectively leverage multi-band information to filter relevant nodes from the rich component-graphs.

Component-graph construction algorithms are in the early development stage as the component-graphs has been introduced not so long ago compared to the classical component-trees. Given the proven richness of the component-graph and the capacity to handle multi-band images, the DAG structures remain challenging for both construction and filtering algorithms.

2.7 Conclusion

This chapter has covered the historical development of connected operators from the early stage on binary images to the extensions on grey-scale and multi-band images. We specifically discuss several advances of connected operators to handle multi-band images, including two major approaches. First, component-graphs fully support multi-band data using partial orderings. Second, the class of MToS and CC-Trees try to address multi-band data with tree-based representations.

While this chapter provides the basis of *connected operators* in mathematical morphology, the next chapter will pay attention to the *component-graph*

structures for multi-band object detection application to astronomical images.

Algorithm 1: The simplified component-graph $\ddot{\Theta}$ construction.

Input : Graph $G = (V, E)$
Input : $\mathbf{F}()$: a function from V to a nonempty set $\mathbb{V}$
Input : *priority*(): a function computes priority value of a node.
Input : *isLeq*(): checks relation $\leq$ between two node levels.
Input : *regionToNode*(): links a region in region-adjacency graph to a node.
Output: The component-graph $\ddot{\Theta}$

```
   /* Compute region-adjacency graph rag                          */
 1 rag = computeRAG(G)
 2 foreach p ∈ rag do
 3 |   regionToNode[p] = 0
 4 |   queue.put(p,priority(F(p)))
   /* Bottom-up propagation                                       */
 5 while not queue.empty() do
 6 |   p = queue.front()
 7 |   if regionToNode[p] == 0 then
   |   |   /* Create a new node                                  */
 8 |   |   regionToNode[p] = makeNode(F(p))
 9 |   |   regionToNode[p].add(p)
10 |   |   fifo.push(p)
11 |   |   while not fifo.empty() do
12 |   |   |   q = fifo.front()
13 |   |   |   foreach neighbor nei of q do
14 |   |   |   |   if isLeq(F(p), F(nei)) then
15 |   |   |   |   |   if F(p) == F(nei) then
   |   |   |   |   |   |   /* nei and p belong to the same node */
16 |   |   |   |   |   |   regionToNode[nei] = regionToNode[p]
17 |   |   |   |   |   else
18 |   |   |   |   |   |   isChild = true
19 |   |   |   |   |   |   foreach father fa of regionToNode[nei] do
20 |   |   |   |   |   |   |   if isLeq(F(p), F(fa)) then
21 |   |   |   |   |   |   |   |   isChild = false
22 |   |   |   |   |   |   if isChild then
   |   |   |   |   |   |   |   /* regionToNode[nei] is a direct
   |   |   |   |   |   |   |      child of regionTonode[p]      */
23 |   |   |   |   |   |   |   regionToNode[p].addChild(regionToNode[nei])
24 |   |   |   |   |   |   fifo.push(nei)
   /* Reorganize nodes and links in a graph array                 */
25 foreach p ∈ regionToNode do
26 |   if regionToNode[p] ≠ 0∧ regionToNode[p].isCanonical then
27 |   |   graph.insert(regionToNode[p])
28 root = addVirtualRoot(gragh)
29 return graph
```

Bibliography

Alonso-González, A., Valero, S., Chanussot, J., Lopez-Martinez, C., and Salembier, P. (2012). Processing multidimensional sar and hyperspectral images with binary partition tree. *Proceedings of the IEEE*, 101(3):723–747.

Aptoula, E. and Lefèvre, S. (2007). A comparative study on multivariate mathematical morphology. *PR*, 40:2914–2929.

Berger, C. and et al. (2007). Effective component tree computation with application to pattern recognition in astronomical imaging. In *IEEE ICIP*, volume 4, pages IV–41.

Breen, E. J. and Jones, R. (1996). Attribute openings, thinnings, and granulometries. *Computer vision and image understanding*, 64(3):377–389.

Carlinet, E. and Géraud, T. (2014). A comparative review of component tree computation algorithms. *TIP*, 23:3885–3895.

Carlinet, E. and Géraud, T. (2015). Mtos: A tree of shapes for multivariate images. *IEEE Transactions on Image Processing*, 24(12):5330–5342.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to algorithms*. MIT press.

Géraud, T., Carlinet, E., Crozet, S., and Najman, L. (2013). A quasi-linear algorithm to compute the tree of shapes of nd images. In *ISMM*, pages 98–110.

Grimaud, M. (1992). New measure of contrast: the dynamics. In *Image Algebra and Morphological Image Processing III*, volume 1769, pages 292–305. International Society for Optics and Photonics.

Grossiord, E., Naegel, B., Talbot, H., Najman, L., and Passat, N. (2019). Shape-based analysis on component-graphs for multivalued image processing. *MMTA*, 3:45–70.

Grossiord, E., Talbot, H., Passat, N., Meignan, M., Tervé, P., and Najman, L. (2015). Hierarchies and shape-space for pet image segmentation. In *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, pages 1118–1121. IEEE.

Klein, J. (1976). Conception et réalisation d'une unité logique pour l'analyse quantitative d'images (ph. d. thesis). *Nancy University, France*.

Kong, T. Y. and Rosenfeld, A. (1989). Digital topology: Introduction and survey. *Computer Vision, Graphics, and Image Processing*, 48(3):357–393.

Kurtz, C., Passat, N., Gancarski, P., and Puissant, A. (2012). Extraction of complex patterns from multiresolution remote sensing images: A hierarchical top-down methodology. *Pattern Recognition*, 45(2):685–706.

Monasse, P. and Guichard, F. (2000). Fast computation of a contrast-invariant image representation. *IEEE TIP*, 9:860–872.

Naegel, B., Passat, B., Boch, N., and Kocher, M. (2007). Segmentation using vector-attribute filters: methodology and application to dermatological imaging. In *ISMM*.

Naegel, B. and Passat, N. (2009). Component-trees and multi-value images: A comparative study. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 261–271. Springer.

Naegel, B. and Passat, N. (2014). Colour image filtering with component-graphs. In *ICPR*, pages 1621–1626.

Naegel, B. and Wendling, L. (2010). A document binarization method based on connected operators. *Pattern Recognition Letters*, 31(11):1251–1259.

Najman, L. and Couprie, M. (2006). Building the component tree in quasi-linear time. *IEEE TIP*, 15:3531–3539.

Najman, L. and Cousty, J. (2014). A graph-based mathematical morphology reader. *Pattern Recognition Letters*, 47:3–17.

Najman, L. and Talbot, H. (2013). *Mathematical morphology: from theory to applications*. John Wiley & Sons.

Oliveras, A. and Salembier, P. (1996). Generalized connected operators. In *Visual Communications and Image Processing'96*, volume 2727, pages 761–772. International Society for Optics and Photonics.

Ouzounis, G. K. and Wilkinson, M. H. (2007). A parallel implementation of the dual-input max-tree algorithm for attribute filtering. In *ISMM (1)*, pages 449–460.

Passat, N. and Naegel, B. (2014). Component-trees and multivalued images: Structural properties. *JMIV*, 49:37–50.

Passat, N., Naegel, B., and Kurtz, C. (2019). Component-graph construction. *JMIV*, 61:798–823.

Perret, B. and Collet, C. (2015). Connected image processing with multivariate attributes: An unsupervised markovian classification approach. *CVIU*, 133:1–14.

Perret, B., Cousty, J., Tankyevych, O., Talbot, H., and Passat, N. (2014). Directed connected operators: Asymmetric hierarchies for image filtering and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 37(6):1162–1176.

Perret, B., Lefevre, S., Collet, C., and Slezak, E. (2010). Connected component trees for multivariate image processing and applications in astronomy. In *ICPR*, pages 4089–4092.

Rosenfeld, A. (1970). Connectivity in digital pictures. *Journal of the ACM (JACM)*, 17(1):146–160.

Salembier, P. (2013). Connected operators based on tree pruning strategies. *Mathematical morphology: from theory to applications*, pages 177–198.

Salembier, P., Oliveras, A., and Garrido, L. (1998). Antiextensive connected operators for image and sequence processing. *IEEE Transactions on Image Processing*, 7(4):555–570.

Salembier, P. and Serra, J. (1995). Flat zones filtering, connected operators, and filters by reconstruction. *IEEE Transactions on image processing*, 4(8):1153–1160.

Salembier, P. and Wilkinson, M. H. (2009). Connected operators. *IEEE Signal Processing Magazine*, 26(6):136–157.

Serra, J. (1982). Image analysis and mathematical morphology. *Academic press*.

Tarjan, R. E. (1975). Efficiency of a good but not linear set union algorithm. *Journal of the ACM (JACM)*, 22(2):215–225.

Urbach, E. R., Roerdink, J. B., and Wilkinson, M. H. (2007). Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):272–285.

Vincent, L. (1993a). Grayscale area openings and closings, their efficient implementation and applications. In *First Workshop on Mathematical Morphology and its Applications to Signal Processing*, pages 22–27.

Vincent, L. (1993b). Morphological grayscale reconstruction in image analysis: applications and efficient algorithms. *IEEE transactions on image processing*, 2(2):176–201.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269.

Wilkinson, M. H. (2011). A fast component-tree algorithm for high dynamic-range images and second generation connectivity. In *2011 18th IEEE International Conference on Image Processing*, pages 1021–1024. IEEE.

Wilkinson, M. H., Gao, H., Hesselink, W. H., Jonker, J.-E., and Meijster, A. (2008). Concurrent computation of attribute filters on shared memory parallel machines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1800–1813.

Wilkinson, M. H. and Westenberg, M. A. (2001). Shape preserving filament enhancement filtering. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 770–777. Springer.

Xu, Y., Géraud, T., and Najman, L. (2012). Morphological filtering in shape spaces: Applications using tree-based image representations. In *ICPR*, pages 485–488.

Xu, Y., Géraud, T., and Najman, L. (2015). Connected filtering on tree-based shape-spaces. *IEEE TPAMI*, 38:1126–1140.

Chapter 3

Multi-band Object Detection with Component-graphs

This chapter introduces a novel morphological approach for object detection in multi-band images relying on component-graphs and statistical hypothesis tests. We first analyze the component-graph capacity at capturing image structures comparing to the classical component-trees. We then introduce two algorithms to filter duplicated and partial nodes in the component-graphs. Experiments demonstrate a significant improvement in detecting objects on both multi-band simulated and real astronomical images.

Contents

3.1 Introduction

This work aims at exploring the use of component-graphs for general object detection. Particularly, we are interested in filtering algorithms on the component-graphs and in the application to the context of astronomical images.

In mathematical morphology, component-trees (CT) and component-graphs (CG) are two classical structures for image modeling and analysis. These structures model images as hierarchical representations using successive thresholding.

The component-trees (Min-Tree, Max-Tree [**?**] [Breen and Jones, 1996], Tree of Shape [Monasse and Guichard, 2000]) benefit from efficient construction and filtering algorithms [Carlinet and Géraud, 2014] [Najman and Couprie, 2006] [Géraud et al., 2013]. They have diverse applications related to connected filtering, object detection, and segmentation, but those are limited to single-band image processing. Extension to multi-band image processing usually requires a total vectorial order (such as lexicographic ordering, reduced ordering), these order relations are application-dependent [Naegel and Passat, 2009] [Perret et al., 2010] [Carlinet and Géraud, 2015].

On the other hand, the component-graph is designed to handle multi-band images by relying on partial orderings [Passat and Naegel, 2014] [Naegel and Passat, 2014]. Beyond the classical multivariate extensions of the component-trees, the component-graph efficiently holds the whole structural information of multi-band images as directed acyclic graph (DAG) variants. Such DAG variants are more general and more powerful at the cost of higher construction and filtering complexities. The component-graph has been increasingly considered for detection and segmentation applications [Grossiord et al., 2019]. This work explores filtering algorithms on the component-graph for object detection.

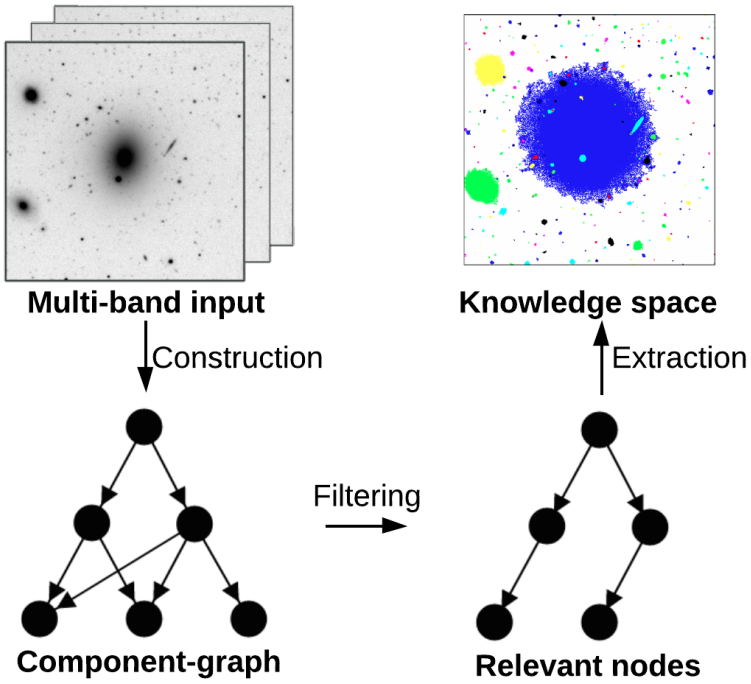


Figure 3.1: CGO filtering method using component-graphs.

In astronomy, the most often used source finder is SExtractor [Bertin and Arnouts, 1996], an efficient and easy-to-use application. However, it breaks down when detecting faint and diffuse objects. For this reason, MTObject/Sourcerer[Teeninga et al., 2016] [Wilkinson et al., 2019] was introduced to improve the SExtractor thresholding strategy by using a component-tree structure. More precisely, MTObject/Sourcerer relies on statistical tests to identify nodes of a Max-Tree that are significantly different from the background. MTObject/Sourcerer has already shown its capability at detecting faint astronomical sources [Haigh et al., 2020] while relying on far fewer parameters than SExtractor. However, both methods focus on single-band processing while most optical astronomical surveys are multi-band. To handle such images, that are expected to lead to increased sensitivity, we propose to generalize the detection method based on statistical testing to the component-graphs. The challenge is to effectively leverage multi-band information to filter relevant nodes from the rich component-graphs.

This chapter proposes a framework to detect the general object on the

component-graph structure. We have experimented with comprehensive analyses on both simulated and real datasets. The overview of the proposed framework is illustrated in Fig. 3.1, it utilizes the component-graph structure to improve object detection sensitivity and to improve object deblending capacity. First, multi-band information improves detection of lower signal-to-noise objects at the same level of confidence (see Fig. 3.2a). Second, the richness of the component-graph helps to deblend overlapping objects that would have been merged with a single band analysis (see Fig. 3.2b). Apart from these advantages, Fig. 2.6 shows that the component-graphs is no longer a tree, but a directed acyclic graph (DAG), which is significantly more challenging to process than the classical component-trees [Grossiord et al., 2019].

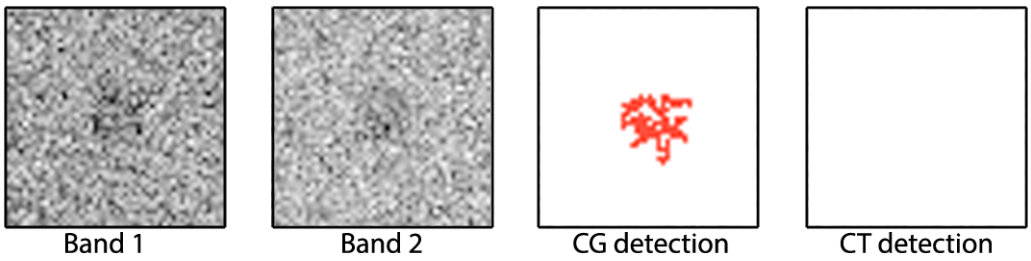
The main contributions of this chapter include:

- Propose a novel multi-band object detection framework relying on component-graphs and application to astronomical source detection.
- Address that the component-graph is better at capturing image structures comparing to classical component-trees.
- Introduce two filtering algorithms to detect duplicated and partial nodes in the component-graphs.
- Improve object detection results on simulated and real multi-band astronomical images.

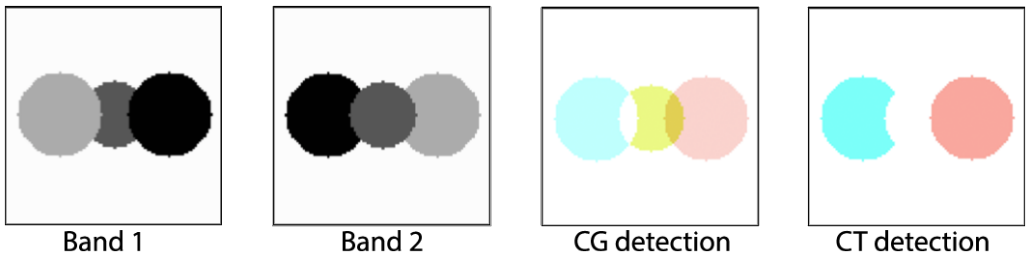
After some preliminary definitions in Sec. 3.2, we introduce CGO in Sec. 3.3 with a set of multi-band node attributes and two methods for duplicated object differentiation and partial node detection. Sec. 3.4 propose an application of CGO to detect sources on astronomical images. Experimental results in Sec. 3.5 show that the proposed approach *CGO* can detect faint sources on simulated and real multi-band images, with significantly better precision and recall than the state-of-the-art method [Haigh et al., 2020] [Teeninga et al., 2016].

3.2 Component-Graphs

This section discusses the simplified component-graph attributes. In-depth presentations of the component-graph and its variants are described in Section 2.2.4, Section 2.2.5 and [Naegel and Passat, 2014; Grossiord et al., 2019].



(a) Object detection sensitivity: A two-band single object input with Gaussian noises and its detection using the component-graph and the component-tree. At the same level of confidence, the component-graph detects one object while the component-tree in separate bands could not capture any object. It shows that the component-graph is able to detect faint sources, i.e., sources at low signal-to-noise ratios using multi-band information.



(b) Object deblending capacity in the component-graph : A two-band input containing three overlapping circles. The middle circle appears in the component-graph as an isolated node while it is merged with adjacent regions in the component-tree of separate bands. This illustrates how the color information present in the component-graph can help to deblend overlapping objects.

Figure 3.2: Illustrations of the component-graph advantages for (a) object detection sensitivity and (b) object deblending capacity.

To be precise, this work uses the simplified version of the CG, denoted $\ddot{\Theta}$ (see Fig. 2.6c) and Section 2.2.5.

Given the component-graph $\ddot{\Theta}$ of the valued graph $(G, \mathbf{F})$, node attributes are essential for node filtering algorithms. Let N be a *node* of the component-graph $\ddot{\Theta}$, we formalize the basic attributes of the component-graph as follows:

- The **level** $\mathbf{L}(\mathrm{N})$ is the infimum of vertex values in the node N.

$$\mathbf{L}(\mathrm{N}) = \bigwedge\{\mathbf{F}(x), x \in \mathrm{N}\}. \tag{3.1}$$

- The **area** $\mathrm{a}(\mathrm{N})$ is the number of vertices belonging to that node.

$$\mathrm{a}(\mathrm{N}) = |\,\mathrm{N}\,|, \text{ i.e., the cardinality of } \mathrm{N}\,. \tag{3.2}$$

- The **parents** $\mathrm{parents}(\mathrm{N})$ are the smallest nodes of the component-graph $\ddot{\Theta}$ larger than the node N.

$$\mathrm{parents}(\mathrm{N}) = \min\{X \in \Psi \mid \mathrm{N} \subsetneq X\}. \tag{3.3}$$

  As a consequence of the partial order relation $\leq$, a node N may have several parent nodes. This leads to the directed acyclic graph structure of the component-graph. In case the order relation $\leq$ becomes a total order, the node is restricted to have a single parent at most, then the graph structure will fall back into the classical tree structure of the component-tree.

- The **significance** $\mathrm{sn}(\mathrm{N}, b)$, $\mathrm{sn}_{\mathrm{syn}}(\mathrm{N})$ and $\mathrm{sn}(\mathrm{N})$ are predicates saying whether the node N is significant respectively in the b-th band, in the synthesized band, and in all bands. The significance attribute should be designed specifically for each application. The synthesized band is a combination of separated bands where the way to combine is also application-dependent. It could be as simple as a summation or averaging. For instance, a measure of eccentricity can be used for elongated object filtering, or compactness can be used for round object detection. Our significance definitions targeting astronomical sources are introduced in Section 3.4.1.

- The **closest significant ancestors** $\mathrm{sn}_{\mathrm{anc}}(\mathrm{N})$ are the smallest significant ancestors of the node N:

$$\mathrm{sn}_{\mathrm{anc}}(\mathrm{N}) = \min\{X \in \Psi \mid \mathrm{N} \subsetneq X \text{ and } \mathrm{sn}(X)\}. \tag{3.4}$$

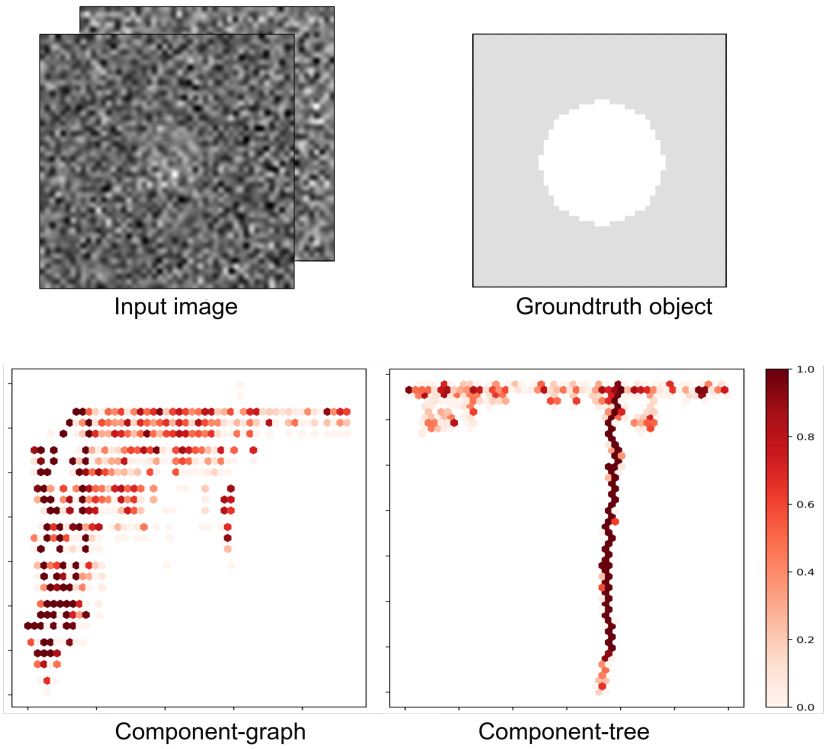


Figure 3.3: Component-tree and component-graph structure differences: (top) A two-band input image containing a single faint source and its ground-truth; (bottom) The component-graph (of the both bands) and the component-tree (of the first band) of the input image where the node color represents the similarity between the ground-truth and the node. Note that the parent relations are not drawn, to simplify the illustrations.

Because of the partial order $\leq$, a node $\mathrm{N} \in \ddot{\Theta}$ may have several closest significant ancestors.

3.3 Filtering the Component-Graph

We introduce CGO (Component-Graph Object), a method to handle multi-band object detection with the component-graph. We first address the transition of object detection from the component-tree to the component-graph, afterward we present the proposed filtering algorithm.

The component-graph is a directed acyclic graph while the component-tree forms a tree. Fig. 3.3 visualizes the structural differences between the component-tree and the component-graph via their similarity maps between each node and the ground-truth node of a single source image. The similarity is measured by the Intersection over Union (IoU) metric, defined as the area of the intersection divided by the area of the union of the two components. In both structures, there exist many candidate nodes (with high IoU scores) associated with the only single object in the input image. For the component-tree, filtering objects from those similar nodes is straightforward, as good candidate nodes of an object form a branch in the tree. On the other hand, the DAG structure of the component-graph allows the candidate nodes to form many branches associated with a single object, which is much more challenging to filter.

We now present a novel algorithm to deal with the multi-band object detection in the component-graph. The main filtering algorithm Alg. 2 takes three inputs: the component-graph $\ddot{\Theta}$ representing the input image; the significance attribute sn() identifying significant nodes; and the function differ() measuring the dissimilarity between nodes. It outputs a list of object nodes. The algorithm is composed of two filtering steps which are described in detail in the two following sections. Intuitively, the first filtering attempts to remove duplicated nodes in the component-graphs. The aim of the second step is then to filter out partial nodes referring to the same object.

3.3.1 Duplicated Object Detection

In the morphological data structures (the component-tree and the component-graph), objects appear differently at different thresholding levels as sequences of significant nodes. For instance, in the case of a single-band input, the object in Fig. 2.5a is represented by the three nodes $\{A, B, D\}$ in the Max-Tree (see Fig. 2.5b). In the case of a multi-band input (see Fig. 3.4a), three nodes $\{B, C, D\}$ in the component-graph $\ddot{\Theta}$ (see Fig. 3.4b) may correspond to one or two objects. Specifying objects among those potentially overlapping nodes is not straightforward on either the component-tree or the component-graph.

In the context of the component-tree with a total order, the function differ() can rely on the main branch assumption [Teeninga et al., 2016]: a node and its main branch node reside in the same object, where the main branch node is defined as the largest significant descendant of a node. A sequence of main branch nodes following a node forms the main branch.

Algorithm 2: Filtering the component-graph $\ddot{\Theta}$

Input : Component-Graph $\ddot{\Theta}$.
Input : Function sn() determines significant node.
Input : Function differ() distinguishes two nodes.
Output: List of object nodes.

`/* Filter duplicated objects */`

foreach *node* $\mathrm{N} \in \{X \in \ddot{\Theta} \mid \mathrm{sn}(X)\}$ *from root to leave* **do**
  if $\mathrm{sn}_{\mathrm{anc}}(\mathrm{N}) = \emptyset$ *or* $\mathrm{differ}(\mathrm{N}, Y)\ \forall Y \in \mathrm{sn}_{\mathrm{anc}}(\mathrm{N})$ **then**
    $\mathrm{objs} \leftarrow \mathrm{objs} \cup \mathrm{N}$

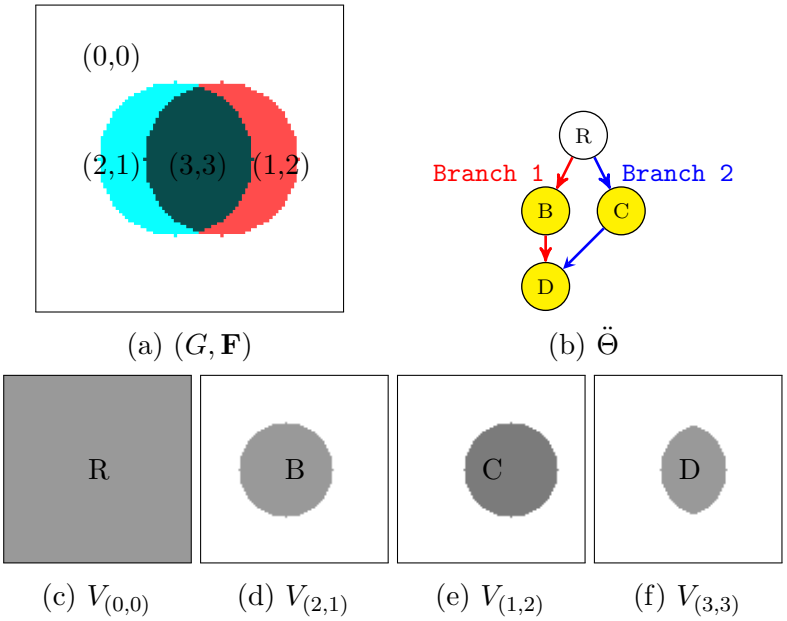
`/* Filter partial nodes */`

foreach *node* $\mathrm{N} \in \mathrm{objs}$ **do**
  if $\mathrm{sn}_{\mathrm{syn}}(\mathrm{N})$ **then continue**
  `/* Function` partial() `details in Alg.` (3) `*/`
  if $\mathrm{partial}(\mathrm{N}, \ddot{\Theta}, b)\ \forall$ *band* b *such that* $\mathrm{sn}(\mathrm{N}, b)$ **then**
    $\mathrm{objs} \leftarrow \mathrm{objs} \setminus \{\mathrm{N}\}$

return objs

Then, all nodes in the main branch represent the same object. Back to the single-band example (in Fig. 2.5a and Fig. 2.5b)), three nodes $\{A, B, D\}$ simply belong to the main branch $(A \rightarrow B \rightarrow D)$ in the Max-Tree, then they all represent a single object.

However, in the context of the component-graph with partial orders, there may exist several branches containing non-comparable nodes belonging to a single object. The main branch assumption is thus not enough to differentiate these branches in the new multi-band context. For example in Fig. 3.4a and Fig. 3.4b, both branches $(B \rightarrow D)$ and $(C \rightarrow D)$ in the component-graph $\ddot{\Theta}$ may correspond to one or two objects. We propose to generalize the main branch approach by using a generic function that measures the dissimilarity between two nodes and that should be designed upon applications. The algorithm Alg. (2) then identifies candidate nodes by browsing significant nodes from the root to the leaves of the component-graph *(line 1)*: If the current significant node does not have any significant ancestor or if it is significantly different (according to the function differ()) from all its significant ancestors then it is an object candidate *(line 2)*; Otherwise, the node is considered a duplicated node. A practical differ() function for astronomical images is described in 3.4.2.



(a) $(G, \mathbf{F})$ (b) $\ddot{\Theta}$

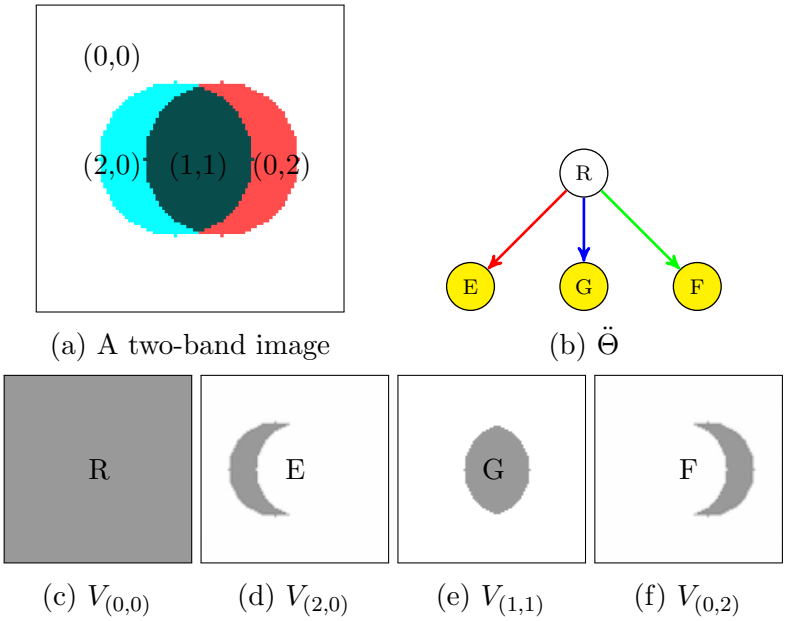
(c) $V_{(0,0)}$ (d) $V_{(2,1)}$ (e) $V_{(1,2)}$ (f) $V_{(3,3)}$

Figure 3.4: Duplicated objects detection: (a) A two-band image, (b) The CG $\ddot{\Theta}$, where significant nodes are marked yellow. `Branch 1` and `Branch 2` are incomparable and growing to the same leaf node.

3.3.2 Partial Node Detection

In the component-graph, significant adjacent nodes can be non-comparable when marginal orders in separate bands disagree. Those nodes can be captured as isolated objects whereas they may belong to the same object. An example is shown in Fig. 3.5, where three significant adjacent nodes E, F, G are non-comparable in a two-band image, but they appear to be detected as three separated objects associated to the three branches $(R, E), (R, F), (R, G)$. Considering the first band, the Max-Tree of the first band is shown in Fig 3.6, nodes E and G should be considered as two parts of a single object, but they are isolated because of the order disagreement in the two-band space of the component-graph. The situation is similar for nodes F and G in the second band.

We propose a partial detection step to validate the significance of the candidates band-by-band to eliminate the partial nodes. The algorithm Alg. (2) checks each candidate node N *(line 4)*: If N is significant in the *synthesized band*, then N is an object node *(line 5)*; Otherwise, the node N is partial



(a) A two-band image (b) $\ddot{\Theta}$

(c) $V_{(0,0)}$ (d) $V_{(2,0)}$ (e) $V_{(1,1)}$ (f) $V_{(0,2)}$

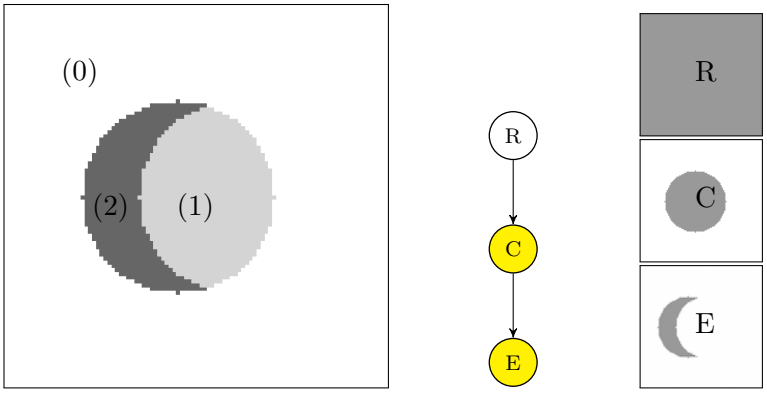
Figure 3.5: Partial object detection: (a) A two-band image I valued on $\mathbb{V} = \{(0,0),(2,0),(0,2),(1,1)\}$ equipped with the marginal order relation $\leq_{\mathrm{m}}$; (b) The CG $\ddot{\Theta}$ of the input image where yellow nodes are significant; and (c-f) The threshold sets V_v for $\boldsymbol{v} \in \mathbb{V}$.

if N is partial in all the bands where N is significant *(line 6-7)* (Alg. (3) determines whether a node is partial in a specific band b).

The idea of Alg. (3) is to test whether N expands any adjacent node of N. For each node N′ adjacent to N *(line 1)*, we look at the Max-Tree of N, N′ and the union U = N ∪ N′, see Fig. 3.7:

- If $\mathbf{L}(\mathrm{N})_b > \mathbf{L}(\mathrm{N}')_b$, N is an isolated significant node in the Max-Tree in band b, then N is an object node regardless of the significance of the union U *(line 2)*, i.e., N is not partial, see Fig. 3.8(a-b).

- If $\mathbf{L}(\mathrm{N})_b \leq \mathbf{L}(\mathrm{N}')_b$, then N is included as a part of the union U in the Max-Tree in the band b, see Fig. 3.8(c-f), there are two possibilities that make the candidate node N be partial: First, the union U is not significant in band b, then N becomes part of the non-significant union U *(line 3)*, see Fig. 3.8(c-d); Second, the union U and N′ are both significant, then N is part of the object node U which is represented by N′ *(line 4)*, see Fig. 3.8(e); Otherwise, N′ is non-significant while



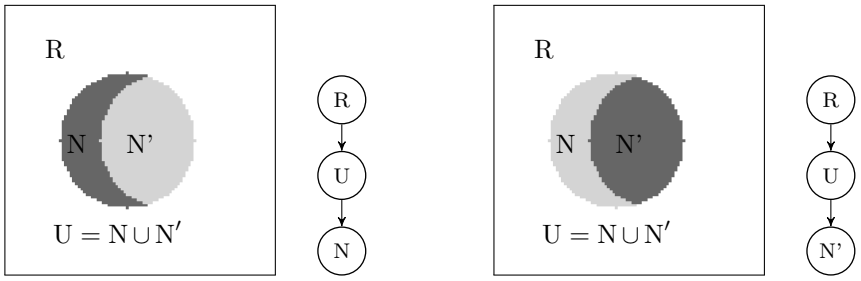
(a) The image first band (b) $\ddot{\Theta}$ (c) V_v

Figure 3.6: Partial object detection: (a) The first band of the two-band image I (see Fig. 3.5) valued on $\mathbb{V} = \{0, 1, 2\}$; (b) The CG $\ddot{\Theta}$ of the input image; and (c) The threshold sets V_v for $\boldsymbol{v} \in \mathbb{V}$.

the union U is significant, then N remains as object node, i.e., N is not partial, see Fig. 3.8(f).

Back to the example in Fig. 3.5 where E, F, G are the three candidate nodes in the graph $\ddot{\Theta}$. The partial detection would validate E as an isolated object because it is significant in the first band and it merges to the root in the second band. Similarly, F is also marked as an isolated object. For G in the first band, it expands into the union $C = G \cup E$: If the union C is significant, then G is part of the significant union C; Otherwise, G is a non-object node. The situation is similar for G and F in the second band. All in all, E and F are object nodes.

Practically, the adjacent nodes $\mathcal{ADJ}(\mathrm{N})$ of the node N are costly to retrieve from the component-graph (at complexity $\mathcal{O}(n^2)$ with n the number of nodes). In this work, we approximate $\mathcal{ADJ}(\mathrm{N})$ by the adjacent sibling set, which is more efficient to retrieve. In the component-graph, the sibling set of a node is reachable in constant time. Then for a node, the adjacency approximation is at complexity $\mathcal{O}(n \cdot k)$ with k the average number of siblings of a node. Because of the complexity constraint ($\mathcal{O}(n^2)$ for fully retrieving adjacent nodes of each node), we are not able to perform a full analysis on the whole dataset to compare the full retrieval and the approximation strategies. However, we have seen the almost identical result on a limited number of sample images. The computation complexity of Alg. (2) and Alg. 3



(a) $\mathbf{L}(\mathrm{N})_b > \mathbf{L}(\mathrm{N}')_b$ (b) $\mathbf{L}(\mathrm{N})_b \leq \mathbf{L}(\mathrm{N}')_b$

Figure 3.7: The Max-Tree of the image containing three nodes: the node N, the adjacent N′ of N, and the union U = N∪N′ in band b.

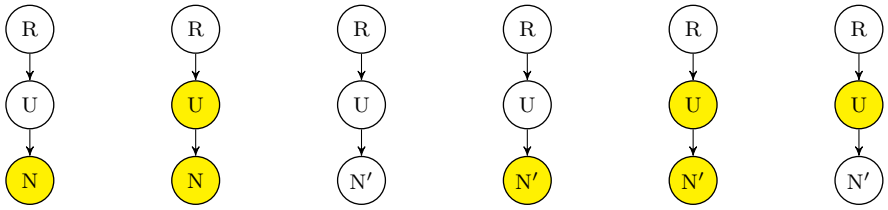


Figure 3.8: The possible links between nodes in the max-tree of N, N′ adjacent to N, and U = N∪N′ in the band b: (a-b) $\mathbf{L}(\mathrm{N})_b > \mathbf{L}(\mathrm{N}')_b$ and (c-f) $\mathbf{L}(\mathrm{N})_b \leq \mathbf{L}(\mathrm{N}')_b$.

is discussed in the following Section. 3.3.3.

3.3.3 Complexity Analysis and Optimization

The time complexity of Alg. 2 is $\mathcal{O}(n \cdot k \cdot m)$, with n the number of nodes in the component-graph, k the average number of siblings of the significant nodes, and m the number of candidate nodes. The number m and k are generally bounded by a low value compared to the number of nodes n.

In detail, the first step *duplicated object detection (line 1-2)* complexity is application dependent, and it can be done optimally in $\mathcal{O}(n)$ if the complexity of the function sn(), the function $\mathrm{sn}_{\mathrm{anc}}()$ and the function differ() is constant. Precisely, the significance and the closest significance ancestors attributes are both pre-computed and stored in the nodes during the component-graph construction, then querying those attributes is $\mathcal{O}(1)$. The function differ()

Algorithm 3: Partial detection: partial($\ddot{\Theta}, \mathrm{N}, b$)

Input : $\ddot{\Theta}$, a component-graph.
Input : $\mathrm{N} \in \ddot{\Theta}$, a candidate node.
Input : b, a significant band.
Output: true if N is a partial node in band b.

```
1 foreach N' ∈ 𝒜𝒟𝒥(N) do
2 |   if L(N)_b > L(N')_b then continue
3 |   if not sn(N ∪ N', b) then return true
4 |   if sn(N ∪ N', b) and sn(N', b) then return true
5 return false
```

can be done in constant time if it relies on the distance between the center pixels which can also be pre-computed during the graph construction.

The complexity of the second step *partial node detection (line 3-4)* is $\mathcal{O}(n^2 \cdot m)$. For each of m relevant nodes found in the first step, the function partial() needs to retrieve adjacent siblings and to check partial conditions for that node at the cost of $\mathcal{O}(n^2)$ for the worst case when the average number of siblings k is closed to n, as mentioned in Alg. 3 section 3.3.2.

All in all, the complexity of the algorithm Alg. 2 is $\mathcal{O}(n^2 \cdot m)$ and the algorithm Alg. 3 is $\mathcal{O}(n^2)$ with respect to n the number of nodes and m the number of relevant nodes in the component-graph.

3.4 Application to astronomical images

We describe an application of the proposed method CGO to detect sources on multi-band astronomical images. As the CGO filtering algorithm requires, we design a significant attribute (sec. 3.4.1) and a node dissimilarity measure (sec. 3.4.2) in the following sections.

3.4.1 Significance Attribute of Astronomical Sources

For astronomical images, we extend the idea of the MTObject significance test [Teeninga et al., 2016] to the multi-band context. This significance measure is based on a chi-square distribution of the brightness of the component pixels. Precisely, the area of the component (i.e., the number of pixels) is the

number of degrees of freedom of the chi-square where each pixel brightness is considered as an independent normal random variable. Its computation relies on two component-attributes: the node normalized power and the node area. Let N be a node in the component-graph $\ddot{\Theta}$.

- The **node power** is the sum of the squared difference between the node pixel values and the level of the parents. Since a node in the component-graph $\ddot{\Theta}$ may have several parents, this definition uses the supremum (average, infimum, max area node can also be used) of the parent levels as a reference:

$$\boldsymbol{\mathcal{E}}(\mathrm{N}) = \sum_{x \in \mathrm{N}} \left(\mathbf{F}(x) - \bigvee_{y \in \mathrm{parents(N)}} \mathbf{L}(y) \right)^{\circ 2}, \tag{3.5}$$

  where $\bigvee$ is supremum operator and $^\circ$ is element-wise power.

  As we can see, the node power is expensive to compute directly by looking at all pixel values lying inside the node. Alternatively, the node power can be computed efficiently via three other intermediate node attributes: the area a(N), the sum of pixel values **sigx**(N), and the sum of squared pixel values **sigxsq**(N):

$$\begin{aligned} \boldsymbol{\mathcal{E}}(\mathrm{N}) = \mathbf{sigxsq}(\mathrm{N}) &+ \mathrm{a(N)}\, \mathbf{F}\left(\mathrm{parents(N)}\right)^{\circ 2} \\ &- 2\, \mathbf{F}\left(\mathrm{parents(N)}\right) \cdot \mathbf{sigx}(\mathrm{N}), \end{aligned} \tag{3.6}$$

  where

  - The sum **sigx** of pixel values belonging to a node is defined as

$$\mathbf{sigx}(\mathrm{N}) = \sum_{x \in \mathrm{N}} \mathbf{F}(x); \tag{3.7}$$

  and where

  - The sum **sigxsq** of squared pixel values belonging to a node is defined as

$$\mathbf{sigxsq} = \sum_{x \in \mathrm{N}} f(x)^{\circ 2}. \tag{3.8}$$

Since the component-graph is not a tree, but a DAG, these three intermediate node attributes can not be accumulated from leaves to root. However, they are still very useful and compact to store. For example in the case of checking partial nodes, the value of parents may change, then node power re-calculation can be done much more efficiently with the intermediate attributes compare to scanning again pixels belong to the node.

- The **node normalized power** normalizes the node power by the local background variance:

$$\boldsymbol{\mathcal{E}'}(\mathrm{N}) = \boldsymbol{\mathcal{E}}(\mathrm{N}) \oslash (\hat{\boldsymbol{\sigma}}^2_{\mathrm{bg}} + \mathbf{L}(\mathrm{parents}(\mathrm{N})) \oslash \mathbf{gain}), \tag{3.9}$$

where

$$\mathbf{gain} = (\hat{\boldsymbol{\mu}}_{\mathrm{bg}} - \bigwedge_{x \in V} \mathbf{F}(x)) \oslash \hat{\boldsymbol{\sigma}}^2_{\mathrm{bg}} \tag{3.10}$$

refers to the CCD gain in astronomy; $\oslash$ is element-wise division; and $\hat{\boldsymbol{\mu}}_{\mathrm{bg}}, \hat{\boldsymbol{\sigma}}_{\mathrm{bg}} \in \mathbb{R}^{\mathrm{c}}$ stand for the mean and the standard deviation of the background of the image $\mathbf{F}$ which has c bands. The background is approximated by the combination of flat tiles which are determined using D'Agostino's K^2 test [Teeninga et al., 2016].

- The **node significance** relies on hypothesis testing. Let b be one band in a c-band image, the definition of the single band significance test $\mathrm{sn}(\mathrm{N}, b)$ is the same as in [Teeninga et al., 2016]:

$$\mathrm{sn}(\mathrm{N}, b) = \boldsymbol{\mathcal{E}'}(\mathrm{N})_b > \mathrm{cdf}\chi^2(\alpha, \mathrm{a}(\mathrm{N})), \tag{3.11}$$

where $\boldsymbol{\mathcal{E}'}(\mathrm{N})_b$ is the normalized node power in band b, $\mathrm{cdf}\chi^2()$ is the chi-square cumulative distribution function, α is a significance level, and $\mathrm{a}(\mathrm{N})$ is the area of the node N. The test is extended to a multi-band significance test $\mathrm{sn}(\mathrm{N})$ defined as follows:

$$\mathrm{sn}(\mathrm{N}) = \Big(\exists b \in [0, c), \mathrm{sn}(\mathrm{N}, b)\Big) \text{ or } \Big(\mathrm{sn}_{\mathrm{syn}}(\mathrm{N})\Big), \tag{3.12}$$

where

$$\mathrm{sn}_{\mathrm{syn}}(\mathrm{N}) = \Big(\sum_{b=0}^{c-1} \boldsymbol{\mathcal{E}'}(\mathrm{N})_b > \mathrm{cdf}\chi^2(\alpha, c\,\mathrm{a}(\mathrm{N}))\Big) \tag{3.13}$$

is the *synthesized band* significance test and c the number of bands.

Intuitively, the node is considered significant if it is so in a single band or in the synthesized band. We aim at leveraging the multi-band information in the synthesized band to detect significant nodes where their signal in separate bands are all non-significant. For checking the separated bands, the first term in Eq. (3.12) guarantees to capture whatever the single-band significance test can capture in each band. For checking the synthesized band, the second term in Eq. (3.12) takes into account the combined power attribute to determine whether the combined signal is statistically significant. Since the multi-band test evaluates all bands simultaneously, it can detect cases where a node is non-significant in all separate bands but is significant in the synthesized band.

3.4.2 Duplicated Astronomical Source Detection

As stated previously in Sec. 3.3.1, objects appear differently as multiple nodes/components at different thresholding levels in the component-trees and component-graphs. These multiple nodes are the main reason leading to multiple detected candidates of a single object. For CGO application to astronomical images, we have to define a function measuring the dissimilarity between two nodes to detect and filter out the multiple nodes.

We observe that the center of astronomical sources/objects is usually brighter and better localized than the outer parts, i.e., the center is more important than the outer parts. In the component-graph, this observation means that two significant nodes with close centers likely represent the same object. Hence, our idea is to measure the dissimilarity of two nodes in the component-graph by comparing the node centers. Formally, we define a predicate differ() expressing whether two nodes in the component-graph belong to the same object as

$$\text{differ}(\mathrm{N}_1, \mathrm{N}_2) = ||\,\text{center}(\mathrm{N}_1) - \text{center}(\mathrm{N}_2)|| < \mathrm{r}, \tag{3.14}$$

where N_1 and N_2 are two nodes in the component-graph $\ddot{\Theta}$, the function center() returns the center pixel of a node, e.g., the brightest pixel of the node, and r is a thresholding radius. The value of r can be determined adaptively by the PSF of the astronomical survey. The center pixel could also be defined as the center of mass or the center of the best fitting ellipse of the node in the component-graph.

3.5 Experiments

This section shows the relevance of our proposed method for object detection in astronomical images. In all the experiments, the graph G is the classic 4-adjacency graph. We compare CGO with the state-of-the-art method MTObject [Haigh et al., 2020] on simulated and real images:

- **Statistical Test Boundaries** (sec. 3.5.1): We investigate the rejection boundaries of the statistical tests on single-band and multi-band components.
- **Detection Capacity** (sec. 3.5.2): This experiment studies how well the object structures are preserved in the component-tree and the component-graph via a simulation.
- **Evaluation on a simulation** (sec. 3.5.3): We compare the detection methods on simulated astronomical images.
- **Evaluation on real images** (sec. 3.5.4): The study assesses the detection methods on real astronomical surveys.

3.5.1 Statistical Test Boundaries

Both CGO and MTObject rely on statistical hypothesis testing to identify significant components. It is important to formalize and visualize the difference between the tests: the single-band test (with respect to MTObject) and the multi-band test (with respect to CGO). If we assume that the noise is Gaussian, then the node normalized power attribute follows a chi-square distribution. At a given significant level α, the rejection boundary for the statistical test is then equal to

$$\mathrm{b}(n) = \Big\{(a,p) \in \mathbb{R}^2 \mid a \in \mathbb{N}, 1 - \mathrm{cdf}\chi^2(n \times a, n \times p) = \alpha\Big\}, \tag{3.15}$$

where n is number of bands; a, p denote the node area and the node normalized power; and $\mathrm{cdf}\chi^2()$ is the chi-square cumulative distribution. Fig. 3.9 and Eq. (3.15) distinctly reveal the theoretical gaps between rejection boundaries, i.e., at the same confidence level, the multi-band statistical test is more sensitive to weak signal than the single-band statistical test. Note that the multi-band gain is not linear to the number of band differences. Generally speaking, the gain of multi-band information statistically benefits the model confidence to determine significant components.

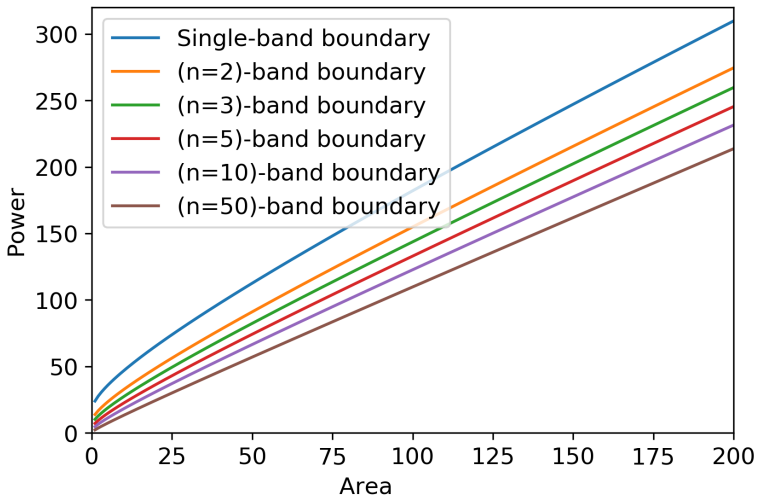


Figure 3.9: Visualization of statistical test rejection boundaries of single-band and multi-band components at the same level of significance $\alpha = 10^{-6}$.

3.5.2 Upper Bound Detection Capacity of the component-tree and the component-graph

To detect target objects, it is critical that the morphological representations of the image must capture them as nodes. In this experiment, we assess how well objects are captured in the component-tree and the component-graph by studying their node *similarity upper bounds* on a synthetic dataset. For a set of nodes $\ddot{\Psi}$ and a ground-truth node gt, we define the similarity upper bound as

$$\mathrm{Sup}(\ddot{\Psi}, gt) = \max_{\mathrm{N} \in \ddot{\Psi}} J(\mathrm{N}, gt), \tag{3.16}$$

where J stands for the Jaccard similarity between two components:

$$J(\mathrm{N}_1, \mathrm{N}_2) = \frac{|\,\mathrm{N}_1 \cap \mathrm{N}_2\,|}{|\,\mathrm{N}_1 \cup \mathrm{N}_2\,|}, \; \mathrm{N}_1, \mathrm{N}_2 \in \ddot{\Theta}\,. \tag{3.17}$$

As we can see, the higher the similarity upper bound, the more likely target objects can be detected and segmented properly. The node associated to the similarity upper bound $\mathrm{Sup}(\ddot{\Psi}, gt)$ can be interpreted as the best object-like node existing in the set $\ddot{\Psi}$.

We analyze the similarity upper bounds of the component-graphs and the component-trees on a *single source simulation* which is shown in Fig. 3.10.

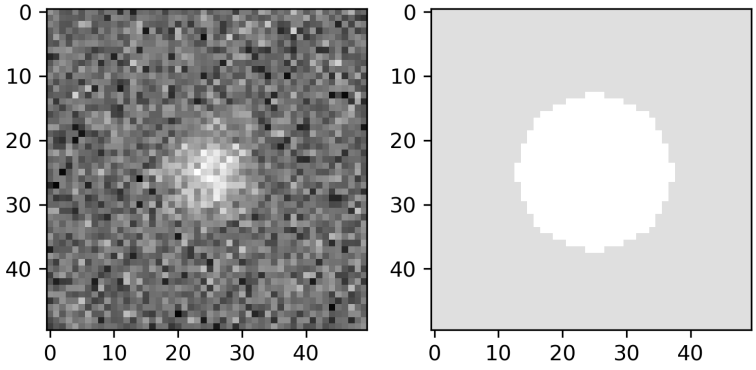


Figure 3.10: *Single Source Simulation*: (left) A synthetic image with $SNR = -0.93$ and (right) the corresponding ground truth.

The simulation includes 10^4 three-band images of size $(50, 50)$ pixels. Each image contains a single point source with Gaussian noise, and the ground-truth is defined as the region covering 99% of the source brightness. The component-trees of the separate bands, the average band, and the component-graph of the three-band image are constructed. Fig. 3.12 shows the average similarity upper bounds with respect to the signal-to-noise ratio of the simulated sources. Visually, Fig. 3.11 illustrates an example of the components which are most similar to the ground-truth captured in the morphological trees and graphs. On this synthetic dataset, the component-graph provides higher similarity upper bounds than the component-tree, i.e., it has better detection capacity comparing to the component-tree.

3.5.3 Evaluation on an Astronomical Simulation

This experiment assesses the detection capacity of the proposed CGO and MTObject on a multi-band simulated dataset. For fair comparisons, we suggest M-MTObject - a straightforward extension of the default single-band MTObject to process multi-band images.

FDS Simulation

This is a three-band simulated astronomical dataset with ground-truth imitating the Fornax Deep Survey [Venhola, 2019] [Venhola et al., 2018], a wide field imaging survey of the Fornax Cluster using ESO's VST telescope. It

contains 1500 stars and 4000 galaxies. Because the component-graph $\ddot{\Theta}$ construction is computationally expensive ($\mathcal{O}(n^2)$), we sliced the simulation into tiles of size (500×500) pixels with overlapping of 250 pixels. For each tile, we have a three-band image with a ground-truth segmentation. The full-size multi-band simulation and ground truth are visualized in Fig. 3.13.

Metric

We use precision, recall, and F1-score, as in [Haigh et al., 2020]. The evaluation matches at most one detected object in the detection map to each target object in the ground-truth map. Each target object in the ground-truth map is represented by its brightest pixel called its *representative pixel*, hence each representative pixel is included in at most one object in the detection map. If a detected object contains several representative pixels of different target objects, then the detected object is associated to the target object with the brightest representative pixel. Precisely,

- True positive detection TPD: is the number of one-to-one mappings matched between detected objects and representative pixels of target objects.
- False positive detection TPD: is the number of objects in the detection map that do not match any target.
- False negative detection FND: is the number of targets that do not match any object in the detection map.

Then the evaluation metrics of precision, recall, and F1-score are formalized as

$$\begin{aligned} precision &= \left(\frac{TPD}{TPD + FPD}\right), \\ recall &= \left(\frac{TPD}{TPD + FND}\right), \\ Fscore &= \left(\frac{2}{recall^{-1} + precision^{-1}}\right). \end{aligned} \tag{3.18}$$

M-MTObject

This work uses MTObject [Haigh et al., 2020] [Teeninga et al., 2016] as the baseline for astronomical source detection. Along with the default single-

band MTObject, we propose a straightforward extension of MTObject to support multi-band images called M-MTObject, where the Max-Tree is computed on the best signal-to-noise ratio band but the attributes and the statistical test use the information from all the bands.

In detail, M-MTObject firstly constructs the Max-Tree of the best signal-to-noise ratio band of the multi-band input image. The filtering strategy is the same as the MTObject statistical test, but node attributes are accumulated from all the bands. Basically, all the bands are forced to follow the selected Max-Tree, i.e., to follow the total order of the best signal-to-noise ratio band. This approach is simple, but each band has its own total order which likely disagrees with the total order of the selected band in some regions. These conflicted regions will introduce false positive significant nodes, causing false positive detection.

Quantitative Results and Discussion

We compare CGO versus the state-of-the-art MTObject/M-MTObject on the three-band FDS Simulation. The common parameter for both methods is the significance level $\alpha \in \{10^{-i} | i \in \mathbb{N} \wedge i \in [3, 100]\}$. Besides, the radius threshold r is another parameter that can be tuned for CGO. However, it can be determined adaptively by the PSF of the input astronomical image survey. Since the signal close to the border of the image is less reliable, we eliminate detected objects whose center is lying within 100 pixels from the borders. Precision and recall curves are presented in Fig. 3.14.

It is clear that our proposed method CGO significantly improves on MTObject at precision and recall metrics in the FDS Simulation. Both CGO and MTObject demonstrate robustness with favorable recalls (> 0.7) given any choice of the parameter α. This is a huge advantage compared to the widely known SExtractor [Bertin and Arnouts, 1996] which has a lot of parameters. Moreover, the precision of the M-MTObject extension drops significantly compared to the others. This can be explained by the inconsistency between the single-band Max-Tree structure and the multi-band attributes.

3.5.4 Evaluation on real astronomical Surveys

We assess CGO versus MTObject/M-MTObject on real multi-band astronomical images SDSS, KiDS, and HST.

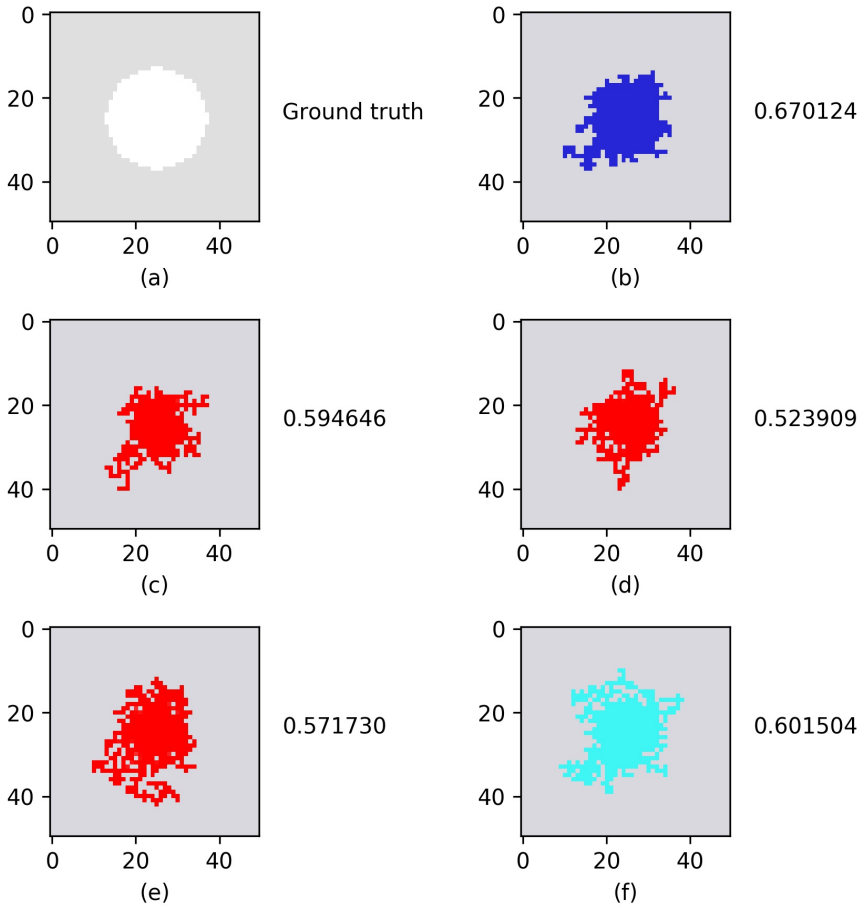


Figure 3.11: *Single Source Simulation* An example of similarity upper bound components Sup with Jaccard index $J(N, gt)$ on: (a) Ground truth component gt; The most similar component to the gt in (b) the component-graph of the multi-band; (c-e) in the component-tree of three separate bands; and (f) in the component-tree of the average band.

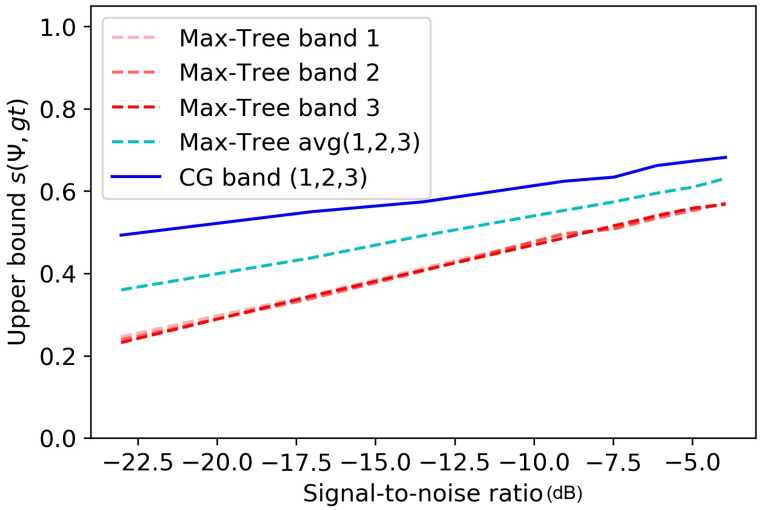


Figure 3.12: Detection upper bounds of the morphological structures.

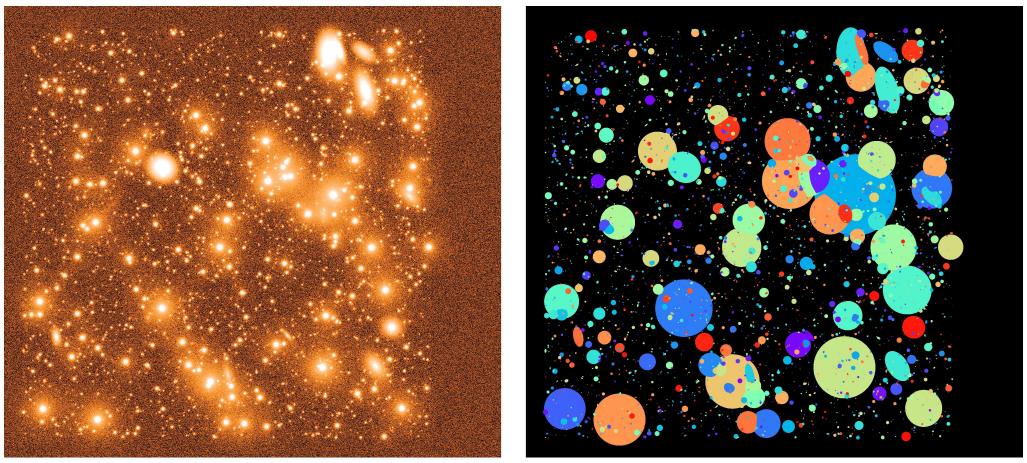


Figure 3.13: FDS simulation: (left) The three-band simulated image and (right) the ground-truth map represents stars/galaxies as separate color blocks.

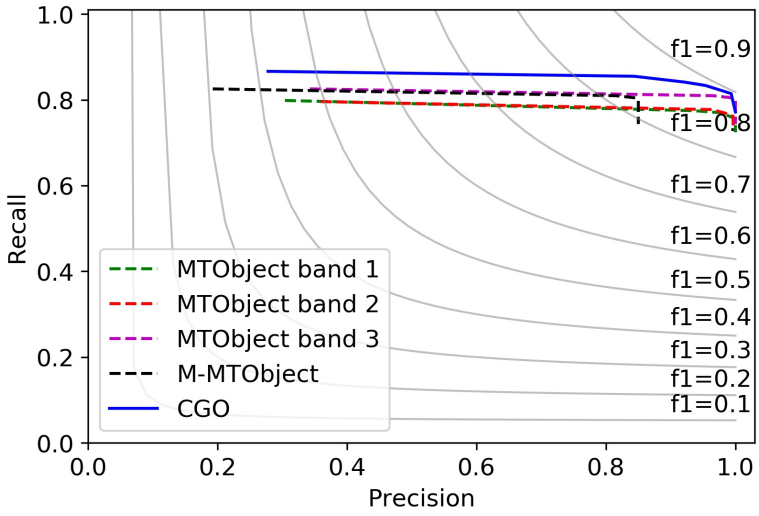
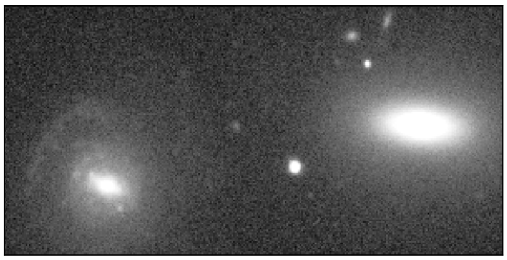
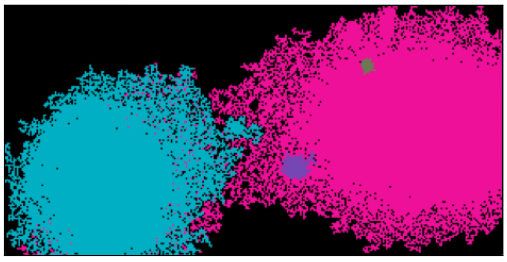


Figure 3.14: Evaluation on the FDS Simulation.



(a) SDSS cutout band g.

(b) CGO result on band (g,r).



(c) MTObject result on band g.

(d) MTObject result on band r.

Figure 3.15: Experiment on a two-band SDSS image.

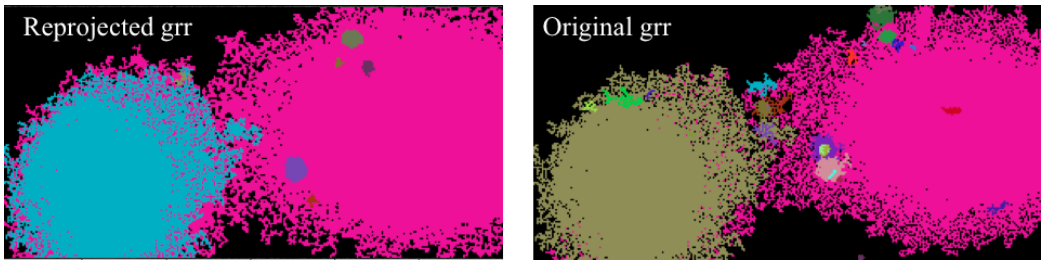


Figure 3.16: CGO detection result of a calibrated multi-band SDSS image and a non-calibrated corresponding (right).

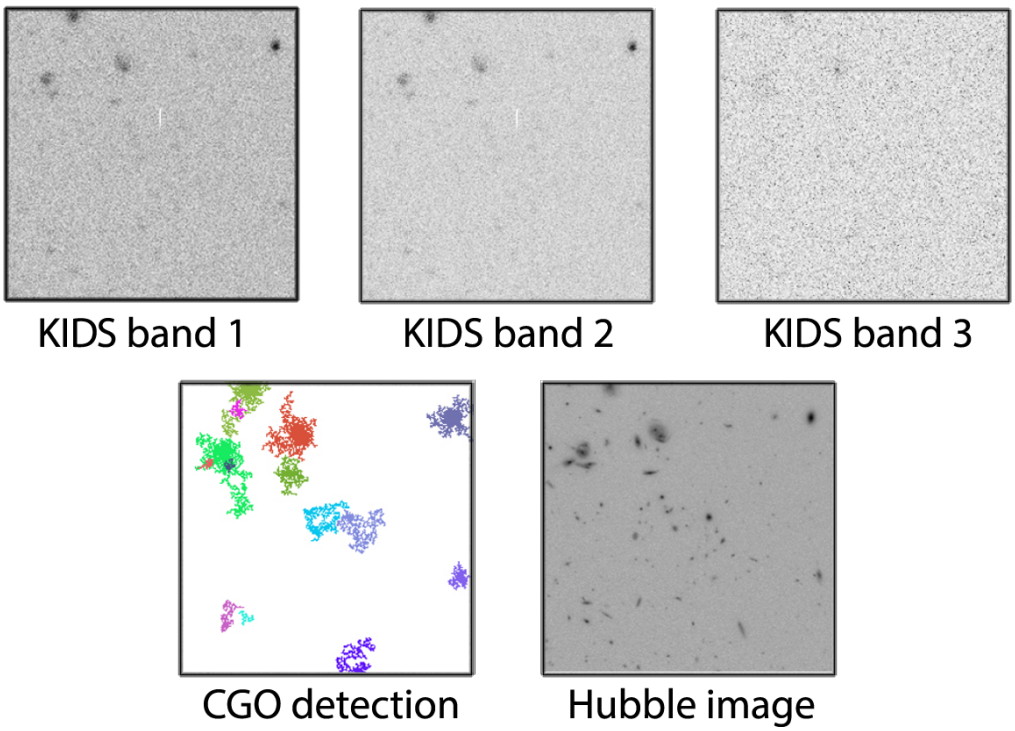


Figure 3.17: KiDS-HST Dataset: A cross-match between a three-band KIDS images and Hubble images.

Real Dataset

We use three astronomical multi-band Surveys: the Sloan Digital Sky Survey (SDSS), the Kilo-Degree Survey (KiDS, [Kuijken et al., 2019]), and the Hubble Space Telescope Cosmic Assembly Near-infrared Deep Extragalactic Legacy Survey (HST CANDELS, [Koekemoer et al., 2011]). The results of CGO and MTObject on the real images are shown in Fig. 3.15, Fig. 3.16, and Fig. 3.17.

For real images, it is critical to guarantee that all bands are registered. As a pre-processing step, multi-band images are re-projected in a single and consistent world coordinate system which is always available in the image metadata. The difference between calibrated and non-calibrated multi-band image detection results are shown in Fig. 3.16, as it can be seen, non-calibrated input causes a lot of unexpected false positives. These false positives are un-aligned records of objects on non-calibrated bands.

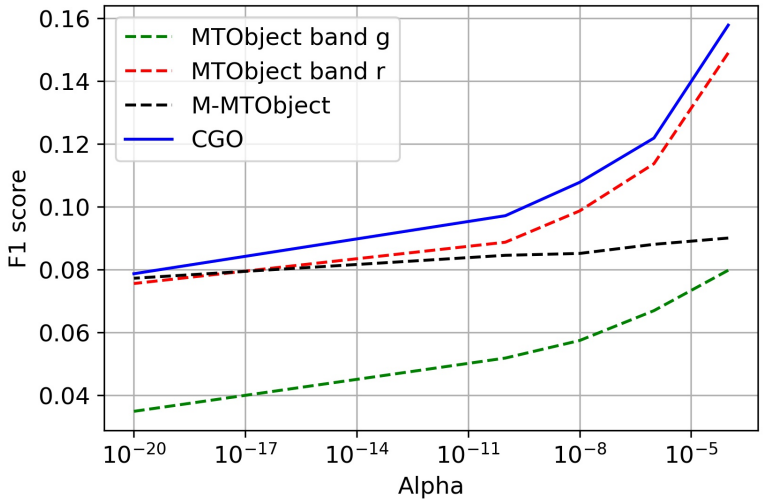


Figure 3.18: Evaluation on the KiDS-HST Dataset.

Ground-truth for CGO

We used 100 image pairs, where each pair consisted of KiDS and HST CANDELS cutouts sharing the same field of view and centered on the same galaxies. All cutouts were taken from the same four source tiles: three KiDS tiles in `u`, `g` and `r` bands, and one HST CANDELS tile observed with the Advanced Camera for Surveys (ACS) in the `F814W` filter. All cutouts were located in RA range $[53.0; 53.2]$ and DEC range $[-27.9; -27.7]$ in the KiDS-South region of the sky. Since HST CANDELS cutouts have much higher resolution and signal-to-noise ratio, we used the detection results obtained with MTObject on these cutouts as the ground-truth for the KiDS images.

Metric

We use the same metrics as mentioned in Sec. 3.5.3.

Quantitative Result and Discussion

We compare CGO and MTObject [Teeninga et al., 2016] on the registered KiDS-HLA images. As shown in Fig. 3.18, CGO achieves a better F1-score than MTObject on this real dataset. These results are consistent with the experiment performed in the FDS Simulation and visual assessment. Note

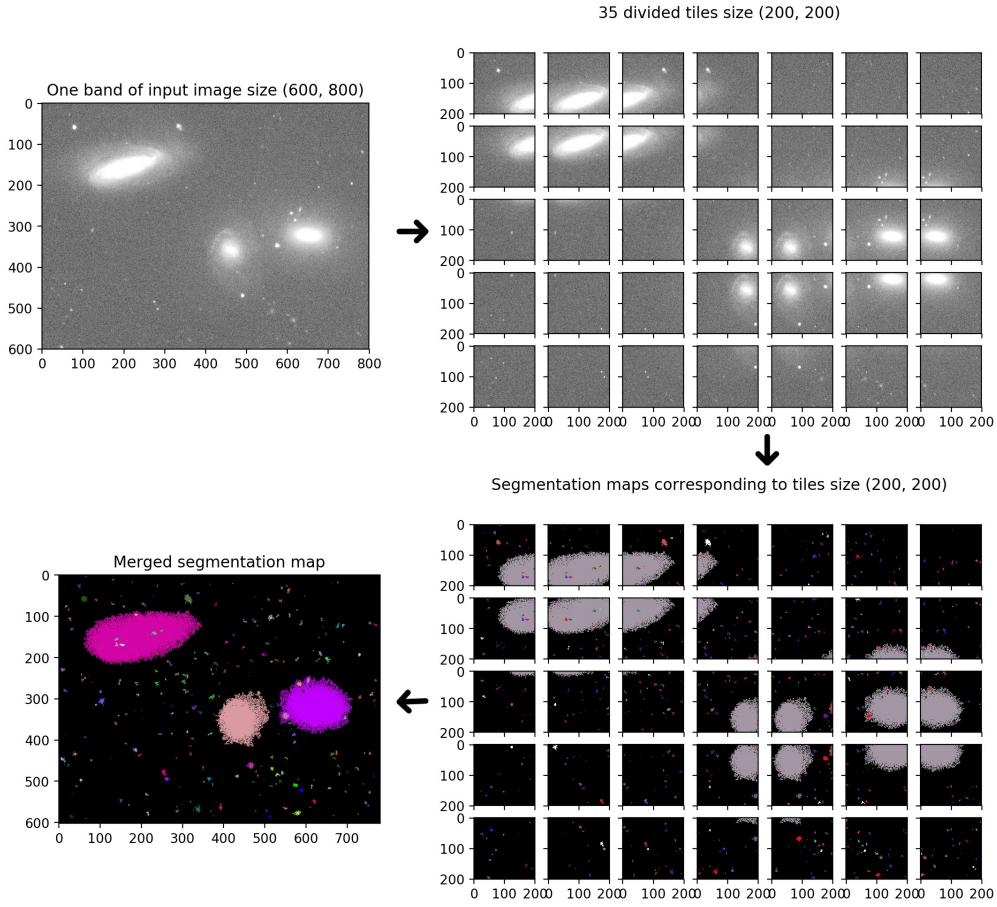


Figure 3.19: A version of paralleled CGO on image partitions: First, input images are sliced into overlapping partitions. Then sub-graphs are constructed in parallel, each sub-graph is computed using the sequential algorithm; Afterward, each sub-graph are filtered independently and merged back into a single detection map.

that all the F1-scores on the KiDS-HLA experiment are much lower than the F1-scores in the FDS Simulation test. The reason is that HLA images (reference images) are much deeper, ie., they contain more detail compared to the KiDS images, therefore many objects in the reference images are just impossible to detect on the KiDS images.

3.6 Conclusion and Perspective

In conclusion, we have explored how the component-graph structures can handle source detection on multi-band data. We proposed CGO – an object detection framework along with a set of novel node attributes on the component-graph, the framework has been applied to detect multi-band astronomical sources [Nguyen et al., 2020, 2021]. Our studies have shown that the component-graphs are better at preserving object structures comparing to the classical component-trees. Comprehensive experiments on both simulation and real astronomical surveys consistently confirm that CGO outperforms state-of-the-art at precision, recall, and F1-score metrics. However, a current limitation of the proposed approach is its time complexity which prevents the processing of large images at once.

While this chapter has been focused on morphological approaches for object detection, the next chapter will turn our attention to machine learning approaches. In particular, we will investigate the class of convolutional neural networks (CNN/ConvNets).

Bibliography

Bertin, E. and Arnouts, S. (1996). Sextractor: Software for source extraction. *Astronomy and Astrophysics Supplement Series*, 117:393–404.

Breen, E. J. and Jones, R. (1996). Attribute openings, thinnings, and granulometries. *Computer vision and image understanding*, 64(3):377–389.

Carlinet, E. and Géraud, T. (2014). A comparative review of component tree computation algorithms. *TIP*, 23:3885–3895.

Carlinet, E. and Géraud, T. (2015). Mtos: A tree of shapes for multivariate images. *IEEE Transactions on Image Processing*, 24(12):5330–5342.

Géraud, T., Carlinet, E., Crozet, S., and Najman, L. (2013). A quasi-linear algorithm to compute the tree of shapes of nd images. In *ISMM*, pages 98–110.

Grossiord, E., Naegel, B., Talbot, H., Najman, L., and Passat, N. (2019). Shape-based analysis on component-graphs for multivalued image processing. *MMTA*, 3:45–70.

Haigh, C., Chamba, N., Venhola, A., Peletier, R., Doorenbos, L., Watkins, M., and Wilkinson, M. (2020). Optimising and comparing source extraction tools using objective segmentation quality criteria. *Astronomy and Astrophysics*.

Koekemoer, A. M., Faber, S., Ferguson, H. C., Grogin, N. A., Kocevski, D. D., Koo, D. C., Lai, K., Lotz, J. M., Lucas, R. A., McGrath, E. J., et al. (2011). Candels: the cosmic assembly near-infrared deep extragalactic legacy survey—the hubble space telescope observations, imaging data products, and mosaics. *The Astrophysical Journal Supplement Series*, 197(2):36.

Kuijken, K., Heymans, C., Dvornik, A., Hildebrandt, H., de Jong, J., Wright, A., Erben, T., Bilicki, M., Giblin, B., Shan, H.-Y., et al. (2019). The fourth data release of the kilo-degree survey: ugri imaging and nine-band optical-ir photometry over 1000 square degrees. *Astronomy & Astrophysics*, 625:A2.

Monasse, P. and Guichard, F. (2000). Fast computation of a contrast-invariant image representation. *IEEE TIP*, 9:860–872.

Naegel, B. and Passat, N. (2009). Component-trees and multi-value images: A comparative study. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 261–271. Springer.

Naegel, B. and Passat, N. (2014). Colour image filtering with component-graphs. In *ICPR*, pages 1621–1626.

Najman, L. and Couprie, M. (2006). Building the component tree in quasi-linear time. *IEEE TIP*, 15:3531–3539.

Nguyen, T., Chierchia, G., Razim, O., Peletier, R., Najman, L., Talbot, H., and Perret, B. (2021). Object detection with component-graphs in multi-band images: Application to source detection in astronomical images. *IEEE Access*, pages 156482–15649.

Nguyen, T. X., Chierchia, G., Najman, L., Venhola, A., Haigh, C., Peletier, R., Wilkinson, M. H. F., Talbot, H., and Perret, B. (2020). Cgo: Multi-band astronomical source detection with component-graphs. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 16–20. IEEE.

Passat, N. and Naegel, B. (2014). Component-trees and multivalued images: Structural properties. *JMIV*, 49:37–50.

Perret, B., Lefevre, S., Collet, C., and Slezak, E. (2010). Connected component trees for multivariate image processing and applications in astronomy. In *ICPR*, pages 4089–4092.

Teeninga, P., Moschini, U., Trager, S. C., and Wilkinson, M. H. (2016). Statistical attribute filtering to detect faint extended astronomical sources. *MMTA*, 1.

Venhola, A. (2019). *Evolution of dwarf galaxies in the Fornax cluster*. PhD thesis, University of Groningen.

Venhola, A., Peletier, R., and et al. (2018). The fornax deep survey with the vst-iv. a size and magnitude limited catalog of dwarf galaxies in the area of the fornax cluster. *Astronomy & Astrophysics*, 620:A165.

Wilkinson, M. H. F., Haigh, C., Gazagnes, S., Teeninga, P., Chamba, N., Nguyen, T. X., Talbot, H., Najman, L., Perret, B., Chierchia, G., Venhola, A., and Peletier, R. (2019). Sourcerer: A robust, multi-scale source extraction tool suitable for faint and diffuse objects. In *IAU 355 Symposium*.

Part II

CONVNET AND MORPHOLOGY FOR ASTRONOMICAL OBJECT DETECTION

Chapter 4

ConvNet Object Detection Literature

In this chapter, we review the methodologies for general object detection in the field of deep learning that later led to the development of the proposed deep learning approach of this thesis. Section 4.1 provides an overview of Convolutional Neural Network-based (ConvNet/CNN) models for visual perception tasks. After that, the following sections specifically narrow down to Region-based Convolutional Neural Networks (R-CNN). In particular, Sec. 4.2 presents the generalization of the R-CNN model while Sec. 4.3, Sec. 4.4, Sec. 4.5, Sec. 4.6, and Sec. 4.7 describe the evolution and essential components of the R-CNN variants.

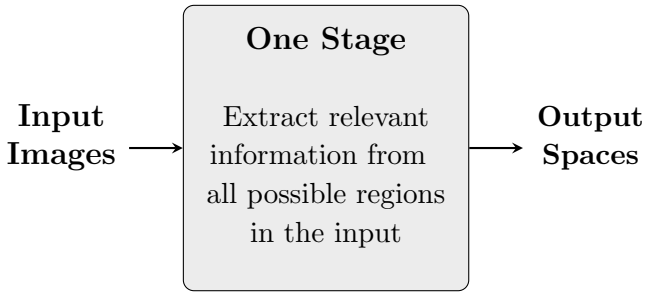
Despite the fact that deep learning is growing rapidly, the model's architectures and core components remain fundamental. This thesis has chosen to go in the direction of the R-CNN models as base architectures to develop our ideas. However, we aim at designing the proposed ideas as independent and transferable components to other classes of ConvNet models.

Contents

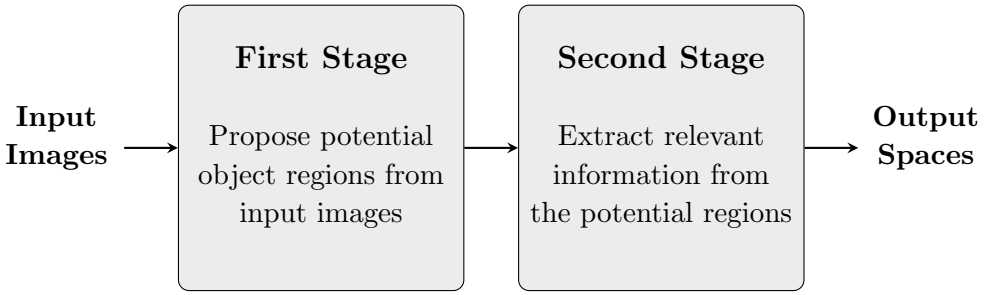
4.1 ConvNet-based Object Detectors

This section reviews the state-of-the-art convolutional neural network-based (ConvNet/CNN) models for object detection. We then explain the reason to focus on Region-based Convolutional Neural Networks (R-CNN) - two-stage instance segmentation models that target to localize, classify, and segment objects on images simultaneously.

In machine learning, the class of ConvNet models has been showing great performances in visual perception tasks such as object detection, classification, segmentation, to name a few. Given the ConvNet-based models' diversity, their key components include feature extractor backbone, prediction heads for specific tasks (localization, classification, and segmentation), preprocessing and post-processing modules. Generally, ConvNet-based object detection models can be grouped into two categories: One-stage detectors and two-stage detectors. The one-stage detectors prioritize inference speed while the two-stage detectors pay more attention to optimize detection accuracy.



(a) The one-stage model overview.



(b) The two-stage model overview.

Figure 4.1: The object detection models overview.

- The one-stage detectors (like OverFeat [Sermanet et al., 2013], YOLO [Redmon et al., 2016], SSD [Liu et al., 2016], RetinaNet [Lin et al., 2017b]) build an end-to-end single deep network composed of convolutional layers, as shown in Fig. 4.1a. The idea of the one-stage detector is to process everything at one pass that helps to speed up the inference and to reduce the number of hyper-parameters. As a consequence of prioritizing speed, the one-stage models suffer from the imbalance of object and non-object proposals in images that influence the detection accuracy. Usually, the non-object proposals outnumber the object proposals, making the model training process biased towards the non-object ones.

- On the other hand, the two-stage detectors (like R-CNN variants [Girshick et al., 2014; Girshick, 2015; Ren et al., 2015; He et al., 2017], SPPNet [He et al., 2015]) consider detection in two stages, shown in Fig. 4.1b. The model proposes a set of proposal regions satisfying

some criterion, in contrast to the one-stage detector sending all possible proposal regions. The first stage also handles the imbalance proposal problem mentioned above before sending proposals to the next step. Then, the second stage extracts relevant information from the proposed regions.

We have chosen to develop our ideas on the R-CNN model for object detection. As the fields of machine learning and deep learning are growing rapidly, state-of-the-art detection models also change quickly. However, we see that the architectures and core components of the model remain fundamental. Such as feature extraction backbones remain essential in any object detectors; also, prediction heads are indispensable, to name a few. At the time of writing this chapter, SCNet [Vu et al., 2021] has recently claimed to outperform the state of the art for object instance segmentation. However, SCNet shares similar model architecture to the general object detection model from the structural point of view - with improvements on box sampling strategies and joint prediction tasks. The two improvements are *Sample Consistency* and *Global Context* in [Vu et al., 2021] respectively. Without the loss of generality, the following sections are dedicated to describing the R-CNN models' whole idea and related components.

4.2 Generalized R-CNN model

In this thesis's scope, we focus on the R-CNN models because we prioritize detection accuracy over inference speed. We first generalize the R-CNN models and then focus on the evolution of the R-CNN variants in the following sections to deeply understand why and how each component in the model was proposed.

As shown in Fig. 4.2, the generalized R-CNN process input image in two stages:

- The First Stage extracts potential region proposals from the whole input images. The stage requires a transformation $b()$ to transform image into feature spaces and a strategy $r()$ to detect potential region proposals containing objects. The proposed regions are then registered by a function $p()$ into the feature space to estimate fixed-sized featurized representations suitable for the next stage. Transformation $b()$ usually prefers convolutional backbones, such as ResNet, VGG, FPN.

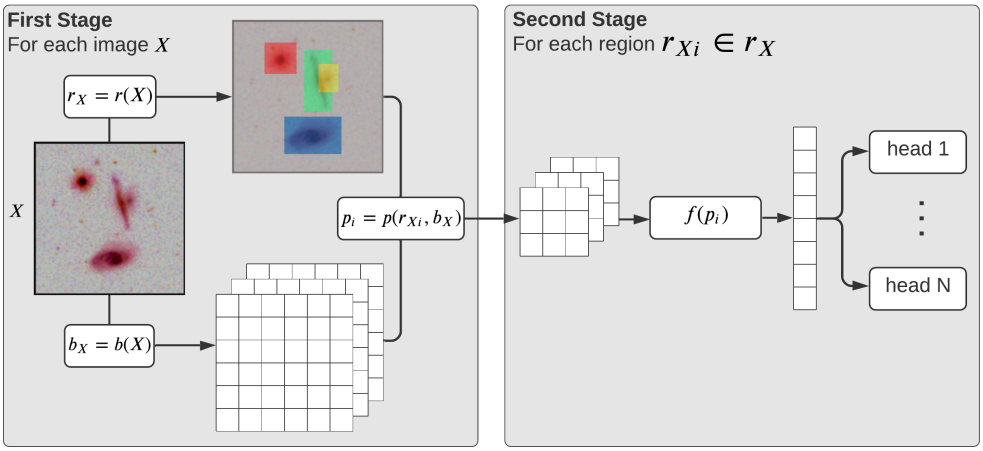


Figure 4.2: Generalized R-CNN Framework

The strategy $r()$ in the first stage can be done with classical image analysis methods (such as Selective Search [Uijlings et al., 2013], EdgeBoxes [Zitnick and Dollár, 2014]) or data-driven method (such as Region Proposal Networks RPN). Both Selective Search and EgdeBoxes are pretty slow comparing to the RPN. Selective Search iteratively merges adjacent image components to propose a hierarchy of candidates based on similarities, while EdgeBoxes uses image contours to rank and predict candidate regions containing objects. Region Proposal Network (RPN) can be trained to generate candidate regions faster and more reliably.

- The Second Stage focuses on each proposed region to extract relevant information. Thanks to the first stage, the input of the second one is very well balanced and normalized. Generally, the stage applies additional processing $f()$ (such as Multi-layer Perceptron MLP or Fully Convolutional Networks FCN) to further reduce the dimension of the region proposal feature, i.e., getting higher abstraction features. Finally, multiple heads (classification and regression) are applied for task-specific predictions, such as class, refined box, key-point, and segmentation mask.

4.3 R-CNN

R-CNN is the initial two-stage detector that uses the Selective Search as a region proposal module and uses another neural network as a detector. The latter includes a feature extractor (a sequence of convolutional layers) followed by a set of linear SVMs and a bounding box regression. Corresponding to the generalized R-CNN framework in Fig. 4.2, we can see the strategy $r()$ uses the Selective Search; the transformation $b()$ transforms input image to itself; the function $p()$ includes cropping and warping; the later processing $f()$ is a convolutional backbone; and the two heads are multi-class SVM and a bounding box regression. Precisely, for each input image:

- Selective Search proposes $N = 2000$ bottom-up candidate regions from input image. Those proposals are category-independent, saying whether it is object or background. Afterward, the proposals are cropped and warped into a fixed-size format and passed into the later stage.
- A convolutional backbone computes a fixed-length feature vector (4096 dimensions) for each proposal. Then, a set of class-specific linear SVMs classifies the feature vectors into categories while another bounding box regression refines the proposals' position.

On object detection datasets PASCAL VOC 2010-12 and ILSVRC 2013, R-CNN out-performed the state-of-the-art OverFeat [Sermanet et al., 2013] back to 2014.

4.3.1 Bounding Box Regression

In order to reduces mislocalizations, a simple bounding box regression method is integrated into R-CNN as a head. Formally, each bounding box is encoded as a set of four parameters: center horizontal/vertical coordinates, width, and height. Given predicted bounding box $p = (p_x, p_y, p_w, p_h)$ and ground truth bounding box $p^* = (p_x^*, p_y^*, p_w^*, p_h^*)$, the box regressor attempts to regress a bounding box $\hat{p}^* = (\hat{p}_x^*, \hat{p}_y^*, \hat{p}_w^*, \hat{p}_h^*)$. To be clear, p is the predicted box from previous Region Proposal module while $\hat{p}^*$ is the predicted box of the Box Regressor. Instead of optimizing the direct box parameters, the Box Regressor [Felzenszwalb et al., 2009] try to regress the bounding box offset $t = (t_x, t_y, t_w, t_h)$ reference to the ground-truth offset $t^* = (t_x^*, t_y^*, t_w^*, t_h^*)$, where

$$
\begin{aligned}
t_x &= \frac{\hat{p}_x^* - p_x}{p_w}, t_y = \frac{\hat{p}_y^* - p_y}{p_h}, \\
t_w &= \log(\frac{\hat{p}_w^*}{p_w}), t_h = \log(\frac{\hat{p}_h^*}{p_h}),
\end{aligned} \tag{4.1}
$$

$$
\begin{aligned}
t_x^* &= \frac{p_x^* - p_x}{p_w}, t_y = \frac{p_y^* - p_y}{p_h}, \\
t_w^* &= \log(\frac{p_w^*}{p_w}), t_h = \log(\frac{p_h^*}{p_h}).
\end{aligned} \tag{4.2}
$$

As we can see, the bounding box offset specifies a scale-invariant translation and log-space height/width shift relative to an object bounding box. To learn the Box Regression, R-CNN optimizes the Sum of Squared Errors (SSE) loss:

$$
L_{reg} = \sum_{i \in \{x,y,w,h\}} (t_i - t_i^*)^2. \tag{4.3}
$$

Limitation

R-CNN is slow in both training and inference comparing to other one-stage detectors. R-CNN suffers significant delay from running Selective Search for each image to propose candidate regions. Extracting feature vectors for each candidate region makes R-CNN even slower. As we can see, each forward pass involves independent modules (Region Proposal, Feature Extractor, Classifier) without sharing computation.

4.4 Fast R-CNN

Fast R-CNN solves the R-CNN speed bottleneck by taking advantage of sharing CNN computation within the model. The main idea comes from the fact that candidate regions in the same image are highly overlapped. Instead of extracting feature vectors independently for each candidate region image (i.e., crop, warp, then compute CNN feature), Fast R-CNN introduces ROI Pooling to extract feature vectors of all candidate regions directly from the CNN feature of the entire image. Since the entire matrix feature is shared, the feature extractor needs only one forward pass to compute the necessary features for all the candidate regions. Technically, the shared feature extractor replaces the last layer of the pre-trained CNN backbone with an ROI Pooling Layer.

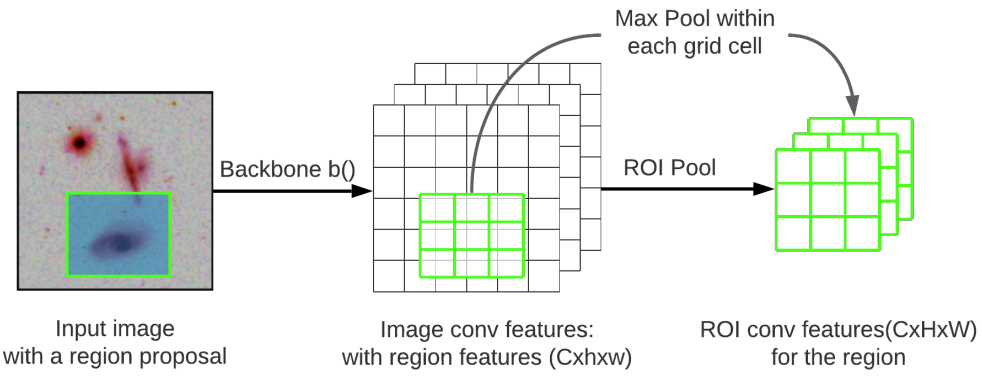


Figure 4.3: ROI Pooling Layer

Then, Fast R-CNN branches two heads for classification and regression.

- A classification head to estimate a discrete probability distribution for each candidate region over the classes. Typically, it is a fully connected layer, followed by a Softmax. This head replaces the linear SVMs in the R-CNN model.

- A regression head to correct the candidate bounding boxes. This head is similar to Box Regression in the R-CNN, except the Fast R-CNN uses Smooth L_1 loss, while R-CNN uses SSE L_2 loss. The Smooth L_1 loss is claimed to be more robust to outliers than the L_2 loss.

4.4.1 ROI Pooling Layer

The ROI Pooling layer uses max pooling to convert features inside any region proposal into a fixed-size feature map. As shown in Fig. 4.3, instead of running a convolutional backbone on the cropped input image, ROI Pooling directly crops feature maps of the input image. In detail, ROI Pooling projects the region proposal onto the image feature maps to obtain the corresponding region feature size $(h \times w)$, which is then divided into an $(H \times W)$ grid. The hyper-parameters H, W are pre-defined for the size of region feature maps. Then, max-pooling selects the maximal value for each cell. The ROI Pooling is applied to each separate band of the region proposal feature

map. The design of ROI Pooling is similar to the pyramid pooling layer in [He et al., 2015].

Note that ROI Pooling is a coarse quantization feature extraction, i.e., both the ROI Pooling projection and grid division approximate the region position to an integer-point location. The approximation trade-off: ROI Pooling is fast and straightforward, but causing a little misalignment between the region and the extracted region features. However, the coarse feature is enough for the two Faster R-CNN heads of box regression and classification. Later, when fine-feature extraction is required, a ROI Align layer will be introduced to replace the ROI Pooling, see Sec. 4.6.2.

4.4.2 Multi-task Loss for Fast R-CNN

Fast R-CNN uses a multi-task loss $\mathcal{L}$ on each labelled candidate region to simultaneously train both the classification head and the bounding box regression head. Let us introduce the following notation:

- i is the index of a candidate region in the mini-batch,
- p_i is the predicted probability of candidate region i over the classes,
- p_i^* is the one-hot encoded ground truth of the candidate region,
- t_i is the predicted bounding box offset,
- t_i^* is the ground-truth bounding box offset.

Then, the loss is defined as

$$\mathcal{L}(p_i, p_i^*, t_i, t_i^*) = \mathcal{L}_{\text{cls}}(p_i, p_i^*) + \lambda \cdot \mathcal{L}_1^{\text{smooth}}(t_i - t_i^*), \tag{4.4}$$

in which

$$\mathcal{L}_{\text{cls}}(p_i, p_i^*) = -p_i^* \log p_i - (1 - p_i^*) \log(1 - p_i), \tag{4.5}$$

and

$$\mathcal{L}_1^{\text{smooth}}(t_i - t_i^*) = \begin{cases} 0.5 \cdot (t_i - t_i^*)^2 & \text{if } |t_i - t_i^*| < 1 \\ (t_i - t_i^*) - 0.5 & \text{otherwise.} \end{cases} \tag{4.6}$$

Thanks to the shared CNN computation, Fast R-CNN is much faster comparing to R-CNN. However, the model is still dependent on the Region Proposal stage, which is computationally expensive.

4.5 Faster R-CNN

Faster R-CNN further unifies the Region Proposal stage into the CNN model by introducing a ConvNet-based Region Proposal Network (RPN). RPN shares the full image convolutional feature with the model, enabling RPN to be trained end-to-end and generate candidate regions nearly cost-free. RPN can be considered as an attention module telling the detector where to look in the image. Faster R-CNN is equivalent to Fast R-CNN plus RPN.

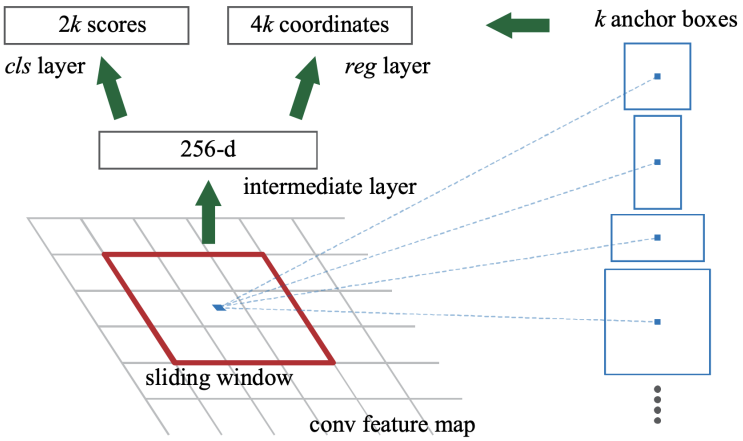


Figure 4.4: Anchor is defined by a center pixel, a scale, and a aspect ratio [Ren et al., 2015].

4.5.1 Multi-Scale Anchors

Multi-scale anchors are designed to handle object detection at different scales. Each anchor is a box that refers to a position. The box is defined by an aspect ratio and a scale. Note that the aspect ratio is respected to the original image. Multi-scale anchors associated to a position consists of multiple anchors at successive scales and different aspect ratios, and they naturally form a pyramid of anchors. For instance, a set of three scales and three aspect ratios yield nine anchors associated to one position. In Faster R-CNN, those multi-scale anchors are generated for each pixel in a feature map. It can be thought of as a pyramid of anchors where each anchor acts as a filter on the feature map. This multi-scale anchor on a single feature map strategy is more

computation and memory cost-efficient than classical approaches based on image pyramid or feature pyramid [Sermanet et al., 2013; Felzenszwalb et al., 2009; He et al., 2015]. Later, the combination of Multi-scale anchors and Feature Pyramid Network [Lin et al., 2017a] will become even more efficient to extract multi-scale feature maps from images, detail in section 4.7.

4.5.2 Region Proposal Network (RPN)

RPN replaces classical region proposal methods (Selective Search, EdgeBox) in the first stage of the two-stage model. This network behaves like an attention mechanism for the Faster R-CNN model. RPN is designed as a neural network, which simply consists of a convolutional layer ($n \times n$ conv filter) followed by a box classification head (1×1 conv filter) and a box regression head (1×1 conv filter). Since RPN takes the full image convolutional feature as input, it is nearly computational cost-free when adding RPN into the R-CNN-style models.

In detail, the input image goes through a backbone network (VGG, ResNet) that outputs convolutional features. The feature extraction step is shared among modules in the whole network, so RPN reuses the output convolutional feature. The first layer ($n \times n$ conv filter) of RPN on top of the feature map can be considered as a sliding window size $n \times n$. A pyramid of multi-scale anchors is then defined for each sliding window by aspect ratios (respect to the original image) and scales (as can be seen in Fig. 4.4). The remaining two heads in the RPN network take sliding window features (i.e., the result of the $n \times n$ conv filter and the input convolutional feature) to perform two tasks for each anchor.

- Classification head predicts an objectness score saying whether it is an object or background region. Till this stage, RPN only cares about object and non-object class, and it acts as a binary classification.

- Box regression head (described in sec. 4.3.1) predicts a parameterized box relative to the anchor.

To train the RPN, anchors will be assigned positive/negative labels, as described in sec. 4.5.1 and sec. 4.5.3. Because the number of background anchors is likely to dominate the number of object anchors, leading to bias toward background anchors, it is advised to optimize the loss on a mini-batch

(256 anchors per image with a balanced negative:positive ratio of 1:1). Let us introduce the following notation.

- M is the mini-batch size to optimize the RPN.
- i is the index of an anchor in the mini-batch.
- p_i is the predicted probability of anchor i being an object.
- p_i^* is the ground truth label (binary) of whether anchor i is an object.
- t_i is the predicted four parameterized coordinates.
- t_i^* is the ground truth coordinates.
- N_{cls} is the normalization term, set to be mini-batch size M (256).
- N_{box} is the normalization term, set to the number of anchor locations (2400).
- λ is the balancing parameter that is insensitive and could be simplified.

Then, the RPN loss is defined as

$$\mathcal{L}_{rpn}\big(p_1, t_1, \ldots, p_M, t_M\big) = \frac{1}{N_{\text{cls}}} \sum_{i=1}^{M} \mathcal{L}_{\text{cls}}(p_i, p_i^*) + \frac{\lambda}{N_{\text{box}}} \sum_{i=1}^{M} p_i^* \cdot L_1^{\text{smooth}}(t_i - t_i^*), \tag{4.7}$$

where

$$\mathcal{L}_{\text{cls}}(p_i, p_i^*) = -p_i^* \log p_i - (1 - p_i^*) \log(1 - p_i).$$

Faster-RCNN heads (classification and regression) select the top-N proposals from the RPN (N=2000 usually). The heads normalize the top-N proposals into fixed-size boxes ROI, then predict multi-class and regress boxes again. The loss is defined in the same manner as the RPN loss where the classification head uses cross-entropy and the regression uses smoothed L_1 loss for only positive boxes.

4.5.3 Non-Maximum Suppression (NMS)

NMS is a post-processing module to get rid of duplicated proposals. The duplicated problem comes from the definition of negative boxes (stands for background regions) and positive boxes (stand for foreground objects) to train the models. Precisely, each box is assigned a similarity score with its corresponding ground-truth box. The similarity is measured by the Intersection over Union (IoU), defined as the area of the intersection divided by the area of the union of the two boxes. A box is considered a positive box if its IoU score with the ground truth box is greater than 0.7. As a consequence, multiple positive boxes can be associated to a single ground truth box. A box is negative if the IoU score with the ground truth is less than 0.3. Models trained on the negative/positive dataset are expected to learn a similar behavior that lists all proposals highly overlapped with the ground truth, i.e., the models accept duplication among the predictions.

To handle the duplicated proposals (or boxes), NMS's idea is to iteratively remove low confidence and highly overlapped proposals. The details are in Algorithm 4, where the given input consist of initial boxes B, corresponding confidence scores S, IoU scores between boxes, and an IoU threshold $\lambda \in [0, 1]$. NMS ranks all the boxes by confidence score. At each iteration, NMS selects the most confidence box and removes highly overlapped boxes which share IoU score greater than the threshold λ with the currently selected box. The procedure continues until all boxes are visited, NMS outputs a list of selected boxes with confidence scores. Literature [Dalal and Triggs, 2005] has shown that NMS does not harm the detection accuracy while significantly eliminating the duplicated detection. NMS has been used in RPN to remove duplicated proposal boxes per anchor level. In Faster R-CNN and Mask R-CNN, NMS remove duplicated predicted object boxes per object class.

4.6 Mask R-CNN

Mask R-CNN conceptually extends Faster R-CNN to instance segmentation context to detect, classify, and generate a mask for each object simultaneously. The Mask R-CNN framework is shown in Fig. 4.5, the major change is adding a parallel branch (i.e., head) to predict object mask along with existing classification and box regression heads. A minor but important change is ROI Align to replace ROI Pooling for coarse region feature extraction.

Algorithm 4: Non Maximum Suppression (NMS)

Input : A set of initial detected boxes $B = \{b_1, \ldots, b_N\}$,
Input : Corresponding confidence scores $S = \{s_1, \ldots, s_N\}$,
Input : IoU threshold $\lambda \in [0, 1]$.
Output: List of filtered boxes F.

$F \leftarrow \{\}$
while $B \neq \{\emptyset\}$ **do**
  `/* Select the most confidence box */`
  $m \leftarrow \arg\max(S)$
  $F \leftarrow F \cup b_m$
  $B \leftarrow B \setminus b_m$
  $S \leftarrow S \setminus s_m$
  `/* Filter out boxes highly overlapped with the selected box */`
  foreach $b_i \in B$ **do**
    if $IoU(b_i, b_m) \geq \lambda$ **then**
      $B \leftarrow B \setminus b_i$
      $S \leftarrow S \setminus s_i$
return F

The following sections explain the Mask R-CNN multi-task loss, the mask branch's network architecture, and the ROI Align references to the generalized R-CNN framework.

4.6.1 Mask head network architecture

The mask head is addressed by a Fully Convolutional Network FCN, as shown in Fig. 4.6. For each region, it takes aligned region feature maps as input and predicts a fixed-size binary mask. We can see that region classification and box regression heads respectively output per-class probability vectors and four-dimensional encoded box offsets, i.e., region feature maps are shortened into lower-dimensional vectors. In contrast, the mask head requires a pixel-to-pixel correspondence to encode and preserve spatial information.

As we can see, the mask head purely relies on *conv* and *deconv*, then the head is lightweight with fewer parameters comparing other approaches utilizing fully connected layers [Pinheiro et al., 2015; Dai et al., 2016]. Ex-

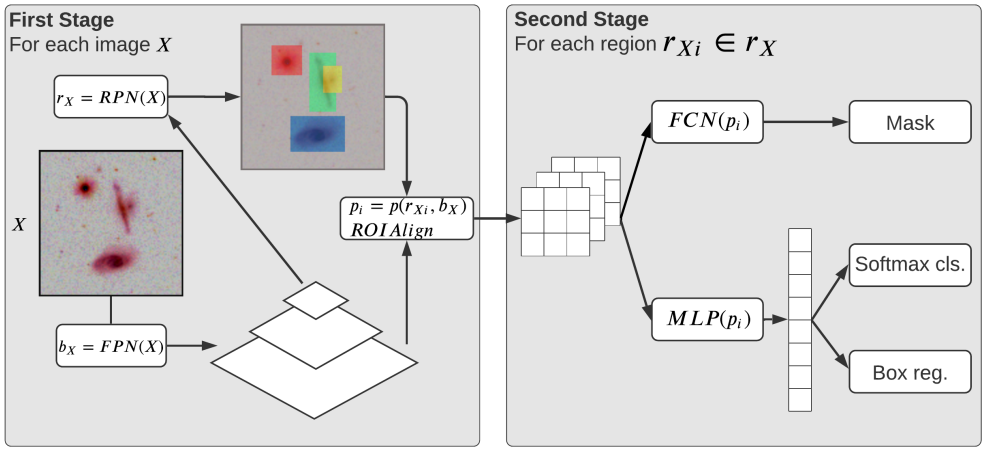


Figure 4.5: Mask R-CNN Framework with FPN backbone.

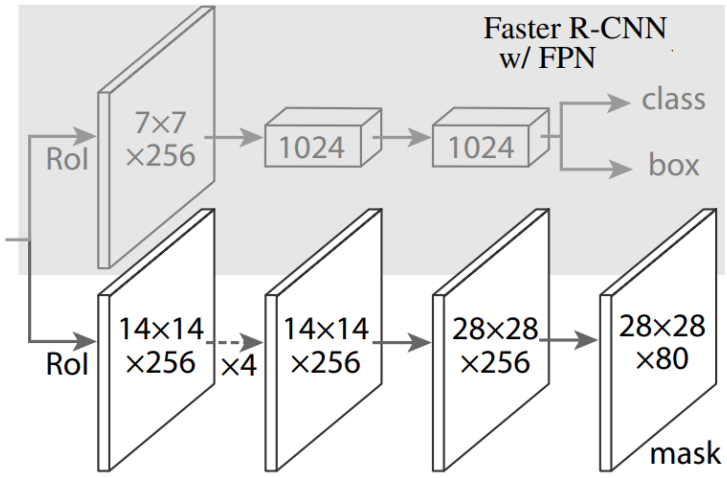


Figure 4.6: The heads of Mask R-CNN with FPN backbone: arrows are either conv, deconv, or fully connect layer; all convs size (3×3) excepts the last conv size (1×1); deconvs size (2×2) with stride 2; and $\times 4$ denotes a stack of four successive convs [He et al., 2017].

perimentally, the mask head adds about 20% overhead to the Faster R-CNN counterpart during inference.

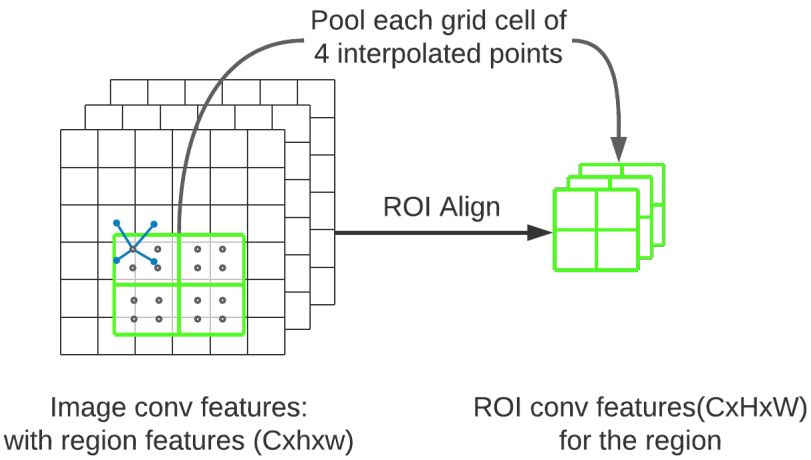


Figure 4.7: ROI Align Layer

4.6.2 ROI Align Layer

ROI Align is a quantization-free feature extraction to obtain fine region features from the full feature maps. In contrast to the regression and classification heads, the mask head maps pixels to pixels between input region features and output masks. Hence, fine and well-aligned feature maps may not impact the classification and regression, but they play an important role in mask prediction.

To address the misalignment, as shown in Fig. 4.7, ROI Align projects region to the exact floating-values region position on the image feature maps, no quantization at the region boundaries. Again, the region is divided into grid cells without quantization. The value of each cell is aggregated (max pool or average pool) from four regularly sampled points. Each point value is interpolated neighboring pixels in the feature maps using bi-linear interpolation [Jaderberg et al., 2015]. ROI Align is simple but effective to align region features, and it has demonstrated essential in the mask head of Mask R-CNN.

4.6.3 Multi-task Loss for Mask R-CNN

Following the generalized R-CNN framework's spirit, Mask R-CNN optimizes the Faster R-CNN heads (box classification and box regression) and the new mask head in parallel. During training, Mask R-CNN defines a multi-task loss as

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{mask}}, \tag{4.8}$$

where $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{reg}}$ are the same as Faster R-CNN losses, defined in Eq. (4.6) and Eq. (4.3). The loss $\mathcal{L}_{mask}$ only includes k_{th} mask if the region is associated with the ground truth class k, yielding

$$\mathcal{L}_{\text{mask}} = -\frac{1}{m^2} \sum_{1 \leq i,j \leq m} \left[y_{ij} \log \hat{y}_{ij}^k + (1 - y_{ij}) \log(1 - \hat{y}_{ij}^k) \right] \tag{4.9}$$

where y_{ij} is label of $cell_{(i,j)}$, i.e., label of the pixel at i^{th} row j^{th} column in the true mask for the region of size $(m \times m)$; and $\hat{y}_{ij}^k$ is the predicted value of the same cell in the mask learned for the ground-truth class k over K classes.

It is important to remark that the mask branch generates K binary masks for each input region, but only one mask associated with the ground truth class contributes to $\mathcal{L}_{mask}$. This design allows the mask branch to generate masks across classes independently, i.e., the mask of a class does not compete with the mask of other classes.

4.7 Feature Pyramid Network FPN

This section recaps the Feature Pyramid Network FPN [Lin et al., 2017a], which is designed to build feature pyramids inside deep convolutional backbones efficiently. We later describe how to adopt FPN in Faster R-CNN/Mask R-CNN for object detection and in RPN for bounding box proposal generation.

In computer vision, feature pyramids are fundamental to deal with object detection at different scales [Dalal and Triggs, 2005; Lowe, 2004]. Objects at different scales are expected to appear similarly at different pyramid levels, i.e., scale-invariant property. Generally, feature pyramids can be obtained by two main approaches:

- (1) Pyramid of images: Feature pyramids can be computed from the image scales independently. This approach is computationally expensive and slow because feature extractors must run on multiple images corresponding to multiple scales. Consequently, computation time and memory are about to increase linearly with the number of pyramid levels.

- (2) Pyramid of feature maps: They construct a feature pyramid directly from a single-scale feature (usually down-sampling the finest resolution feature of the best image scale or taking advantage of the convolutional backbones, such as FPN) for faster computation.

In the era of deep networks, hand-crafted features are gradually replaced by deep convolutional networks serving as backbone feature extractors [Krizhevsky et al., 2017]. The layer-by-layer architecture of the convolutional backbones naturally computes image feature hierarchy corresponding to multi-scale levels. Single Shot Detector has tried to leverage these hierarchies to generate a pyramid of feature maps at almost cost-free in one pass [Liu et al., 2016]. However, there are still semantic gaps between different scales at each depth.

Feature Pyramid Network FPN [Lin et al., 2017a] further takes advantage of the convolutional backbone hierarchy to generate a feature pyramid with strong semantic at all scales. The FPN architecture is shown in Fig. 4.8. To enhance semantic abstraction between levels, FPN defines top-down pathway and lateral connection to combines low-resolution features (correspond to strong semantic layers) with high-resolution features (correspond to weak semantic layers). The FPN strategy outputs rich semantic pyramid features while still building quickly on a single-scale image.

As it can be seen from Fig. 4.8, the bottom-up pathway is the forward convolution of the backbone that generates fine-to-coarse features layer-by-layer. On the other hand, the top-down pathway generates coarse-to-fine features with the help of the lateral connection. At each level, the lateral connection takes two steps:

- up-sample the higher semantic feature map by a factor of two,

- merge the up-sampled feature with the corresponding bottom-up feature map using element-wise addition.

To note that, similar works also adopt the top-down and skip connections [Ghiasi and Fowlkes, 2016; Honari et al., 2016] to produce a single

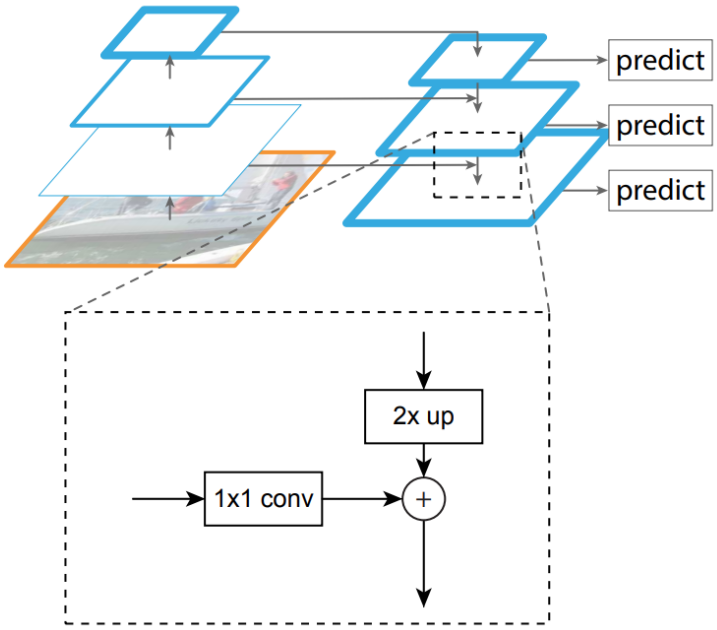


Figure 4.8: The FPN bottom-up/top-down pathway architecture with lateral connections: $(2 \times up)$ is up-sampling by a factor of 2; $(1 \times 1conv)$ is convolution operator to reduce channel dimension but retain the spatial resolution; and merge operator $(+)$ is element-wise addition [Lin et al., 2017a].

feature map at a fine-resolution ultimately. In contrast, the FPN spirit is to generate a feature pyramid with strong semantic at all levels.

4.7.1 Feature Pyramid Network for RPN

RPN is a simple convolutional network relying on multi-scale anchors on top of a single-scale feature map. The pre-defined multi-scale anchors help the network to cover objects of different shapes and scales. The combination of FPN and RPN is expected to gain both extraction speed and feature semantic. To do so, [He et al., 2015; Lin et al., 2017a] replaced the single-scale feature map by multiple feature maps from the FPN feature pyramid. Multi-scale anchors on a single feature map are turned into single-scale anchors on each feature map of the feature pyramid. The anchors are the same, but they are now mapping to the appropriate feature map level in the feature

pyramid. The combination of RPN and FPN only changes the way to extract features, so the loss and training remain the same as the original RPN.

For instance, in a RPN with a ResNet backbone, $\{P_3, P_4, P_5\}$ denote the three feature maps of the FPN, corresponding to the output of three last residual blocks $\{C_3, C_4, C_5\}$ of the ResNet. RPN on a single feature map C_3 would define multi-scale anchors by three sizes $\{32, 64, 128\}$ and three aspect ratios $\{1:2, 1:1, 2:1\}$, i.e., nine anchors at each sliding window position over the C_3 feature map. In the case of RPN with FPN on the same ResNet backbone, aspect ratios remain the same, but three sizes $\{32, 64, 128\}$ are associated to the three feature maps $\{P_3, P_4, P_5\}$ respectively. In total, there are still nine anchors at each sliding window position over the pyramid.

4.7.2 Feature Pyramid Network for R-CNN Variants

R-CNN variants, including Fast/Faster/Mask R-CNN, use deep convolutional backbones as feature extractors. So, theoretically, those backbones can be replaced by the FPN without affecting the whole network training.

In case of the single-scale feature map, Fast/Faster R-CNN use ROI pooling while Mask R-CNN uses ROI Align to extract feature for a region proposal over the whole feature map. To adapt with FPN, i.e., with multiple feature maps at different pyramid levels, [He et al., 2015; Lin et al., 2017a] assigned region proposals to appropriate pyramid levels. Formally, a region proposal size $(w \times h)$ is assigned to feature map at pyramid level P_k with

$$k = \left\lfloor k_0 + \log_2 \left(\frac{\sqrt{w \times h}}{224} \right) \right\rfloor, \tag{4.10}$$

where k_0 is the target level corresponds to a region proposal size (224×224) and 224 is the canonical ImageNet pre-training size that should be changed according to the pre-training dataset. Intuitively, smaller region proposals are mapped to lower pyramid level (or finer-resolution feature) and similarly larger ones are assigned to higher pyramid level (or coarser-resolution feature).

4.8 Conclusion

This chapter has provided an overview of object detection models based on Convolutional Neural Networks (ConvNet/CNN). Besides, we specifically

describe the idea and evolution of the R-CNN variants. We aim at giving an in-depth presentation of each R-CNN variant with related components, such as Bounding Box Regression, NMS, Multi-scale Anchor, Mask Head, or Pyramid Network. Even though this chapter has decided to go in the R-CNN direction, we think that the object detection model's architectures and core components remain fundamental and transferable to other classes of ConvNet models.

While this chapter has provided the basis of ConvNet object detectors, the next chapter will propose two ConvNet-related approaches: One model relies on the R-CNN architecture and the other hybrid model takes the advantages of both morphology and ConvNet.

Bibliography

Dai, J., He, K., and Sun, J. (2016). Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3150–3158.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2009). Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645.

Ghiasi, G. and Fowlkes, C. C. (2016). Laplacian pyramid reconstruction and refinement for semantic segmentation. In *European conference on computer vision*, pages 519–534. Springer.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). 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.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916.

Honari, S., Yosinski, J., Vincent, P., and Pal, C. (2016). Recombinator networks: Learning coarse-to-fine feature aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5743–5752.

Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. (2015). Spatial transformer networks. *arXiv preprint arXiv:1506.02025*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017a). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollar, P. (2017b). Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

Pinheiro, P. O., Collobert, R., and Dollar, P. (2015). Learning to segment object candidates. *Advances in neural information processing systems*, 28:1990–1998.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

Uijlings, J. R., Van De Sande, K. E., Gevers, T., and Smeulders, A. W. (2013). Selective search for object recognition. *International journal of computer vision*, 104(2):154–171.

Vu, T., Kang, H., and Yoo, C. D. (2021). Scnet: Training inference sample consistency for instance segmentation. *AAAI*.

Zitnick, C. L. and Dollár, P. (2014). Edge boxes: Locating object proposals from edges. In *European conference on computer vision*, pages 391–405. Springer.

Chapter 5

ConvNet and Morphology for Astronomical Object Detection

This chapter explores the use of Region-Based Convolutional Neural Network (R-CNN) to tackle object detection in multi-band astronomical images. We begin with an introduction to astronomical object detection in Sec. 5.1 with our motivation to use R-CNN. Afterward, Sec. 5.2 introduces a pipeline to acquires a novel Real KiDS Dataset of multi-band astronomical images. Sec. 5.3 and Sec. 5.4 respectively propose two astronomical source detection models: an R-CNN-based model and a hybrid model that takes the advantages of both morphological-based and machine learning-based models to adapt to astronomical contexts. Finally, experiments in Sec. 5.5 and ablation studies in Sec. 5.6 demonstrate our proposed models gain significant improvements in detecting objects on both multi-band simulated and real astronomical images.

Contents

5.1 Introduction

The research to date has tended to use mathematical morphology to address the source extraction for astronomical images. To name a few, SExtractor [Bertin and Arnouts, 1996], MTObject/Sourcerer [Teeninga et al., 2016] [Wilkinson et al., 2019], and CGO [Nguyen et al., 2020, 2021] are popular and currently being used by the astronomer community. On the other hand, some ConvNet-based models are recently introduced to detect astronomical objects, such as Morpheus [Hausen and Robertson, 2020], Mask Galaxy[Farias et al., 2020], and Astro R-CNN [Burke et al., 2019].

The morphological-based source finders have a long historical development with standard use programs. They are easy-to-use but perform poorly on object segmentation tasks. Additionally, most morphological-based source finders are designed for single-band processing. To date, our proposed CGO (described in Chapter 3 and published as [Nguyen et al., 2020, 2021]) is the only method that intentionally supports multi-band images using the

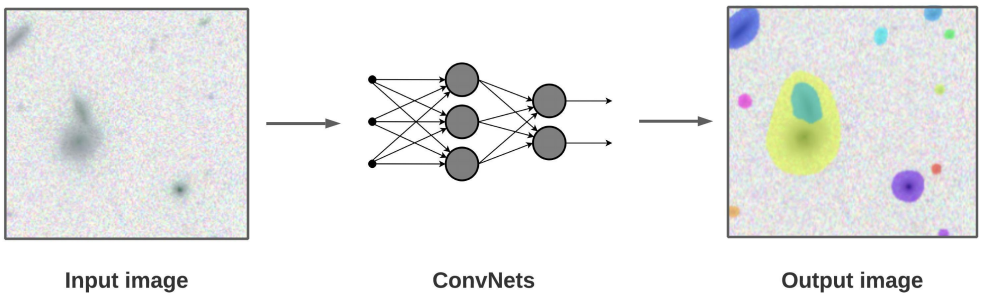


Figure 5.1: The overview of the proposed ConvNet-based object detector.

component-graph structure. CGO uses the multi-band gain to detect fainter objects but at the cost of much higher construction complexity.

Despite being developed recently, the ConvNet-based tools [Hausen and Robertson, 2020; Farias et al., 2020; Burke et al., 2019] have shown potential results comparing to the morphological-based tools. However, these tools are roughly limited to applying computer vision models into astronomical datasets. The main approaches include U-net models for semantic segmentation and R-CNN models for instance segmentation. Two main limitations of ConvNet-based source finders are the lack of standard astronomical datasets and the need to integrate astronomical contexts into models. Given the robustness of ConvNet-based models on astronomical images with recent positive results, the use of these models is still far from the practical needs of astronomers. We believe there is still room not just to apply these models but to re-design the model architecture to adapt to the astronomical context. See Chapter 1 for comprehensive reviews of existing source finders.

State of the art While ConvNet-based source finders are still far from practical usages, a comparison [Haigh et al., 2020] has shown that MTObject achieves the highest scores on both area and detection measures among morphological-based tools (excluding CGO). On the other hand, CGO [Nguyen et al., 2020, 2021] has demonstrated better capacity at detecting faint objects than MTObject but at the cost of higher complexity. In this chapter, we use MTObject and CGO as baselines for comparison.

Motivations To fill the gap of existing methodologies, we aim to approach ConvNet/CNN models to tackle the instance segmentation, as shown in Fig. 5.1. For visual perception tasks, CNN models have shown excellent results in classification, localization, and segmentation.

First and foremost, we target using CNN architectures to naturally take advantage of the multi-band information gain, i.e., CNN models could help encode multi-band images into feature vectors. We not only apply these base models but also tailor the base models with characteristics of astronomical objects and astronomical images.

Second, without constraining object masks (i.e., segmentation) to the thresholded morphological components, we have some degree of freedom to define and optimize CNN-based models that allow overlapping segmentation. In that case, each object segmentation can be modeled as a probabilistic mask. Furthermore, CNN-based models theoretically give some useful information, such as detected class probability or soft segmentation (each pixel has a probability belonging to a mask).

Generally, CNN-based object detection models can be grouped into two categories: one-stage detectors and two-stage detectors. The one-stage detectors prioritize inference speed while the two-stage detectors pay more attention to optimize detection accuracy [Huang et al., 2017]. In this thesis's scope, we choose to focus on the two-stage approaches because we prioritize detection accuracy over inference speed. Despite focusing on the two-stage models, we keep our proposed approaches transferable to other CNN-based models.

The two-stage detectors (like R-CNN variants [Girshick et al., 2014; Girshick, 2015; Ren et al., 2015; He et al., 2017], SPPNet [He et al., 2015]) considers detection in two stages, shown in Fig. 4.1b:

First Stage selectively proposes a set of potential region proposals from the whole input images. During training, the first stage also balances negative and positive proposals before sending them to the next stage. Usually, the first stage can be done with classical image analysis methods (such as Selective Search [Uijlings et al., 2013], EdgeBoxes [Zitnick and Dollár, 2014]) or data-driven methods (such as Region Proposal Networks RPN). Region Proposal Network (RPN) can be trained to generate candidate regions faster and more reliable than the hierarchical-based Selective Search and EgdeBoxes.

Second Stage focuses on each proposed region to extract relevant information. Thanks to the first stage, the second stage's input is well balanced and ready for training. Generally, the stage applies additional processing (such as

Multi-layer Perceptron MLP or Fully Convolutional Networks FCN) further to reduce the dimension of the region proposal features (i.e., getting a higher abstraction feature). Finally, multiple heads (classification and regression) predict task-specific information, such as class, refined box, key-point, and segmentation mask.

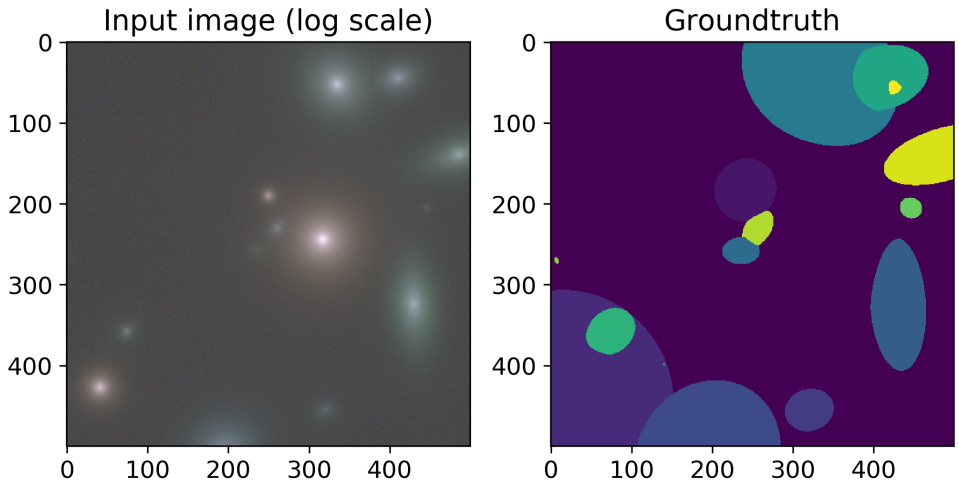
Contributions The main contributions of this chapter include:

- We introduced a real dataset of multi-band astronomical objects with annotated ground-truth in sec. 5.2. The idea to use higher quality images to annotate lower quality images semi-automatically.

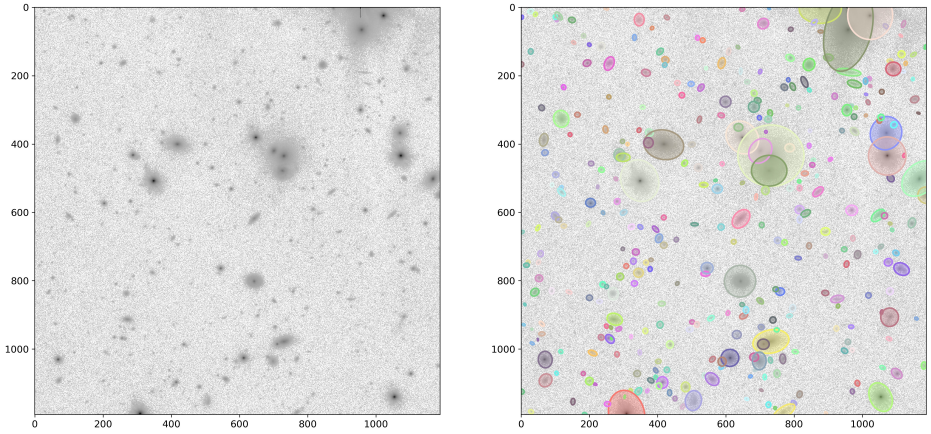
- We proposed an RCNN-based model tailoring object detection on astronomical images (in sec. 5.3). The novelties of the proposed model consist of: a trainable normalization layer that can be trained end-to-end with the whole model, in sec. 5.3.1; CC-NMS module is designed to replace the default NMS at removing multiple detections of a single object, in sec. 5.3.2; and a smoothness regularizer for the segmentation head in the model, in sec. 5.3.3.

- We investigated a hybrid approach leveraging both morphological trees and R-CNN models for object detection (in sec. 5.4). Intuitively, the hybrid model uses a morphological tree to detect potential regions in the first stage, then using convolutional heads to predict relevant information such as labels and segmentation masks.

Note that we have decided to develop a complete approach based on the R-CNN models, but the proposed ideas are transferable to other ConvNet-based models. As machine learning and deep learning are moving rapidly, we think it is important to keep the approach highly adaptable to state-of-the-art models. In particular, the proposed normalization layer is independent of the base model's choice. On the other hand, the CC-NMS (Connected Component NMS) module can apply to any model to avoid duplicated predictions. Finally, the tree-based proposal module in the hybrid approach is also highly compatible with ConvNet-based models.



(a) Simulation: (left) The three-band simulated image and (right) the ground-truth map representing stars/galaxies.



(b) Real images: The KiDS images (left) and annotations (right).

Figure 5.2: Astronomical datasets: Annotations are represented as separated color blocks.

5.2 Astronomical Datasets

This chapter covers datasets being used in this work. In addition to the *FDS Simulation* [Venhola, 2019], we introduce a *Real KiDS Dataset* of multi-band astronomical images. The objects on the real KiDS images are annotated semi-automatically, described in Sec. 5.2.2.

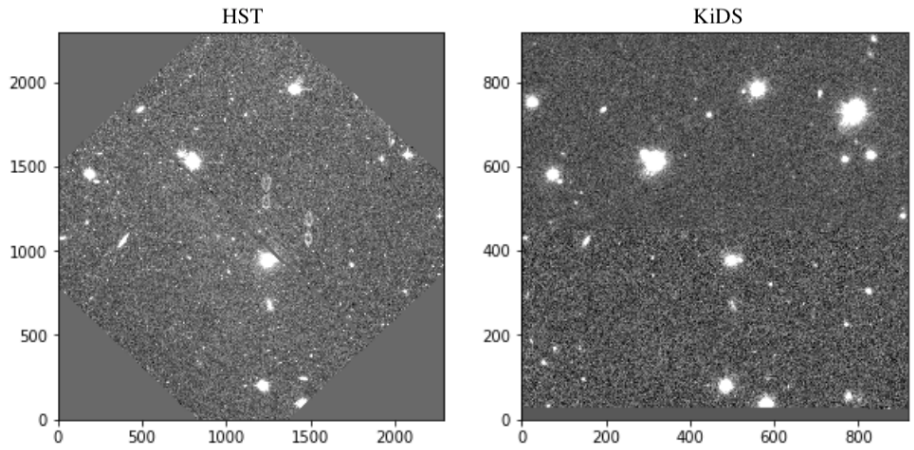


Figure 5.3: A cross-matched HST image (filter F606W) and KiDS image (band r).

5.2.1 FDS Simulation

FDS Simulation simulates three-band astronomical images with ground-truth imitating the Fornax Deep Survey [Venhola, 2019] [Venhola et al., 2018], a wide field imaging survey of the Fornax Cluster using ESO's VST telescope, shown in Fig. 5.2a. This simulation uses the OmegaCAM PSF model, Poisson, and Gaussian noises. It contains 1500 stars as point sources, 4000 background galaxies, and 50 background clusters.

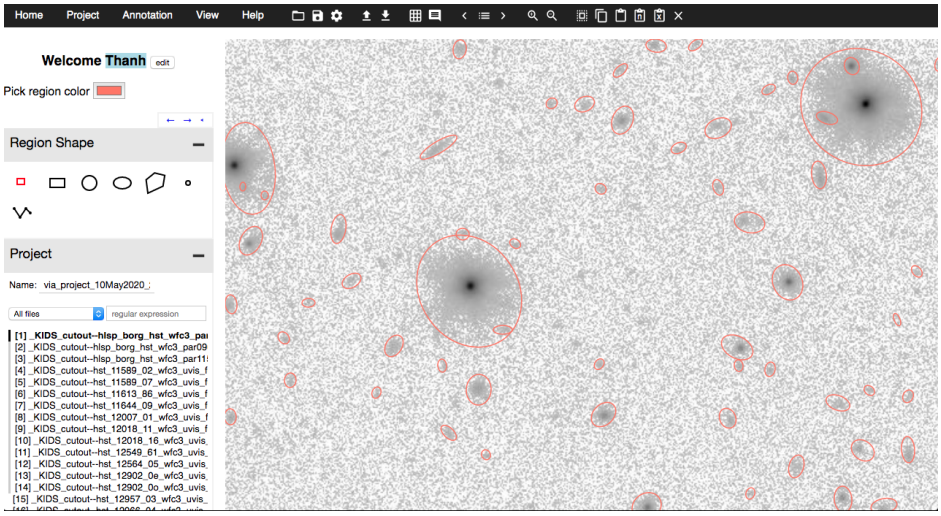
For training purpose, the full simulation is sliced into non-overlapping tiles (size $(512, 512)$ pixels). So for each tile, we have a three-band image (g, r, i) and a corresponding ground-truth mask. A tile is visualized in Fig. 5.2a.

5.2.2 Real KiDS Images

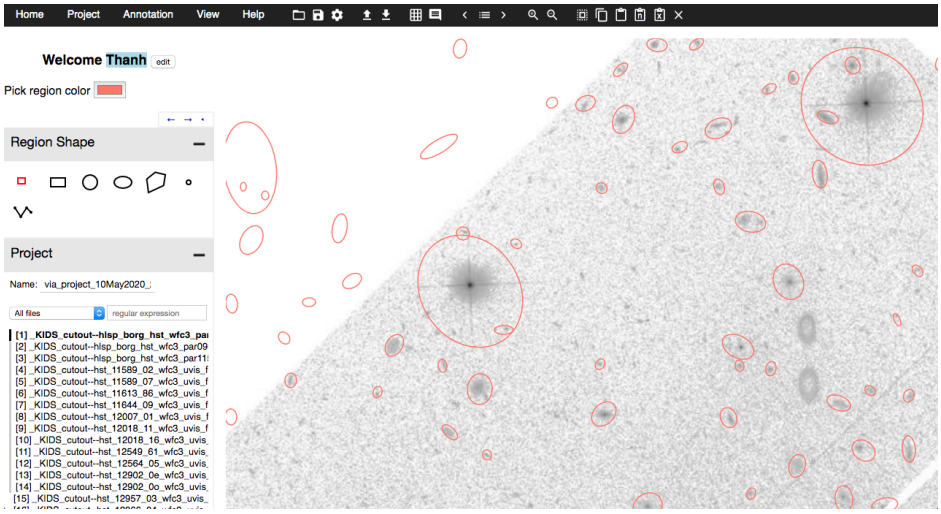
For a real astronomical dataset, we use four-band images from the Kilo-Degree Survey KiDS [Kuijken et al., 2019]. In contrast to the simulation, there is no ground-truth for the real KiDS images, then we set up a semi-auto pipeline to annotate them, and results are shown in Fig. 5.2b. The real dataset along with customized annotation tools are publicly available [*].

The idea is to use high-quality reference images to manually correct automated detections on lower-quality images. The lower-quality images are the

[*] https://github.com/hetpin/sky_imview



(a) Pre-annotated objects on a KiDS image are loaded into Astro Annotator.



(b) The same pre-annotated objects are visualized on the registered HST image.

Figure 5.4: Astro Annotator supports to load, view, and correct the pre-annotated object on registered KiDS and HST images: Each pre-annotated object is approximated by a fitting ellipse.

real KiDS images, while the references are the images sharing the same field of view taken from the Hubble Space Telescope Cosmic Assembly Near-infrared Deep Extra-galactic Legacy Survey [Hubble, 2000; Koekemoer et al., 2011] (HST). Since the HST images have a much higher resolution and signal-to-noise ratio than the KiDS images, we can correct the pre-annotated objects with more confidence. Also, astronomical objects may shine differently at different wavelengths, then it is critical to cross-match KiDS and Hubble images at similar filters/bands corresponding to similar wavelengths, as shown in Fig. 5.3.

Given the KiDS images and the reference HST images, objects are firstly extracted from the KiDS images using existing automated source finders, such as Sourcerer, MTO, and CGO [Teeninga et al., 2016; Wilkinson et al., 2019; Nguyen et al., 2020, 2021]. Second, the pre-annotated objects enter a manual correction supported by higher quality HST images.

To support the second step, we have developed Astro Annotator - a labelling tool to support the manual correction process, based on the easy-to-use VGG Image Annotator (VIA) [Dutta et al., 2016; Dutta and Zisserman, 2019]. The tool supports users to load, view, and correct the pre-annotated objects on both KiDS and HST images simultaneously, as shown in Fig. 5.4. Object segmentation is usually encoded as binary masks or polygons, which are efficient to process but difficult for annotation. Because of that, we practically propose to approximate the segmentation of the pre-annotated astronomical objects by fitting ellipses in the labelling tool. As the nature of astronomical objects with round shapes, the ellipse fits well with the astronomical context. In the case of unusual objects, the labelling tool still supports polygons annotation. Users can consequently correct the approximated ellipses efficiently and export the final segmentation.

The KiDS dataset covers a large sky area, but the number of cross-matched pairs between KiDS and HST is limited. In detail, we have identified and annotated 30 pairs of KiSD-HST images. The dataset includes about 9000 objects with KiDS-image sizes varies from (750×750) to (1800×1800). To increase the amount of data, we have used several augmentation techniques, including random rotation, random cropping, gamma correction, and the combination of them.

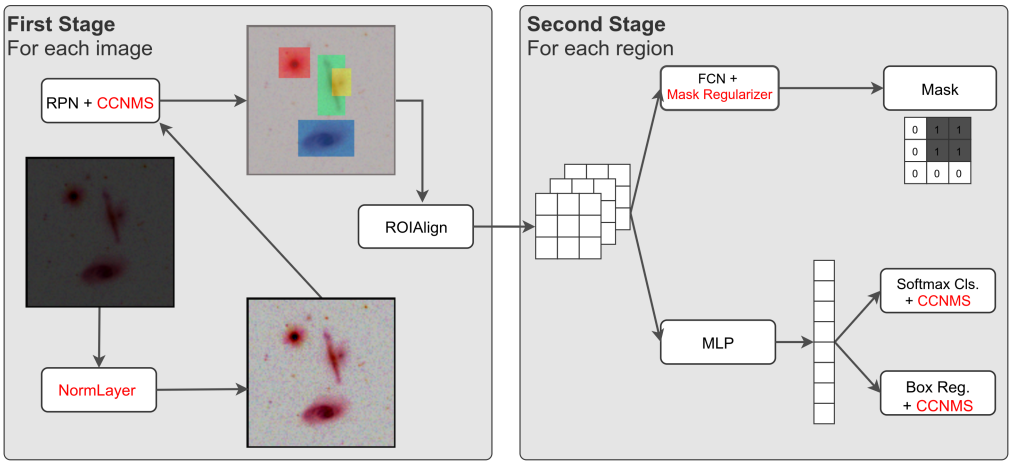


Figure 5.5: The proposed R-CNN model for astronomical object detection: Three novel modules, including a NormLayer, a CC-NMS module, and a Mask Regularizer are red-highlighted.

5.3 Proposed ConvNet Approach

In this section, we propose an R-CNN model to detect objects that adapted to the context of astronomical sources. The overview is presented in Fig. 5.5. The main contributions of the proposed model include:

- We introduce a trainable normalization layer to learn normalization parameters automatically.
- CC-NMS module is designed as a post-processing step to handle duplicated objects in astronomical contexts.
- The mask head is regularized to obtain softened masks for astronomical sources.

5.3.1 Normalization layer

Data normalization is essential, but this step is usually hand-crafted before training. In this work, we propose a normalization layer that can be injected and trained end-to-end together with the whole model.

Generally, image feature extractors (i.e., backbones) are pre-trained with well-known datasets of natural images, such as ImageNet and COCO. The pre-trained weights serve as a good initialization for the later backbone fine-tuning. The closer distribution of the new images compared to the pre-trained images, the easier the fine-tuning is. However, astronomical images and natural images are very different in terms of range and scale. In other words, the new astronomical dataset's distribution is different from the distribution of the pre-trained datasets. To deal with these differences, the general approach is to normalize image values in the new dataset. The normalization parameters are usually hand-crafted or selected via hyperparameter searches which take time and adds bias toward the chosen parameters. In contrast, we design a normalization layer that can be learned automatically with the whole model.

The normalization layer is defined as a sequence of differentiable normalization operators. Particularly in this work, gamma correction and clipping operators are injected directly into our model via the normalization layer. In astronomical images, the majority of pixel values distributes close to the background level while a few bright-object pixels hold extreme values. As shown in Fig. 5.6 (left), the bright-objects affect the upper bound's image range, making the remaining regions look almost flat. These flat regions contain faint features of faint objects and the bright-object surrounding, which are challenging to detect. Hence, we propose to use a clipping operator to expose the faint features of the image, i.e., we focus on the range-of-interest near the background level. Afterward, we propose to use gamma correction to further expose the faint features with a non-linear value mapping function. An advantage of the gamma correction is that it maps the unit range (i.e., range zero to one) to precisely the unit range.

As can be seen in Eq. (5.1) and Eq. (5.2), the gamma and clipping operators are differentiable, and then it allows the model to update the operator parameters via back-propagation. The combination of both operators formalizes the normalization layer, see Eq. (5.3).

- Gamma operator: Given input signal $x > 0$ and gamma correction coefficient $\alpha > 0$,

$$g(x, \alpha) = x^{\alpha}. \tag{5.1}$$

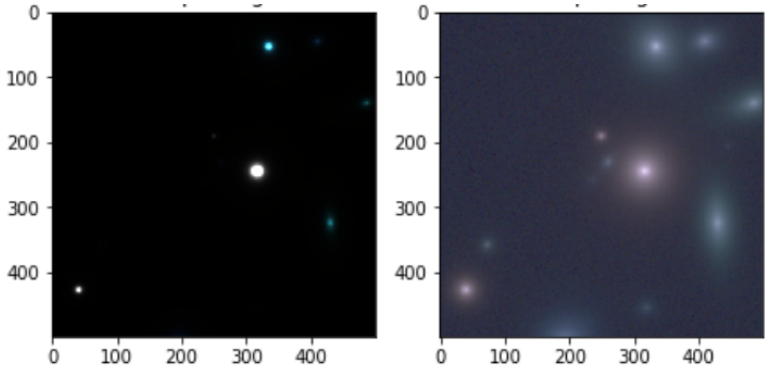


Figure 5.6: Normalization layer: (left) input in log scale and (right) output.

- Clipping operator: Given input signal x and clipping parameter β,

$$c(x, \beta) = \begin{cases} x & \text{if } x < \beta, \\ \beta & \text{otherwise} \end{cases} \tag{5.2}$$

- Normalization layer: Given input signal $x > 0$, gamma correction $\alpha > 0$, and clipping parameter β, the normalization layer is defined as

$$normlayer(x) = g\Big(c(x, \beta), \alpha\Big). \tag{5.3}$$

Intuitively, Fig. 5.6 shows an example of the input and the output of the normalization layer. As we can see, many features in the input are exposed clearly in the output. Besides, the layer also gives a good visualization for astronomical images instead of manual processing.

5.3.2 Duplication Removal Module CC-NMS

We design a post-processing module, called CC-NMS, to improve the NMS module [Dalal and Triggs, 2005; Felzenszwalb et al., 2009] at removing duplicated detection in astronomical images. NMS is widely used to remove duplicated proposals/predictions designed for objects in natural images, described in Sec. 4.5.3. For astronomical context, we observe that the center of astronomical sources is usually brighter and better localized than the outer parts. This observation means that neighbouring detected objects sharing similar centers likely represent the same object. Relying on this observation,

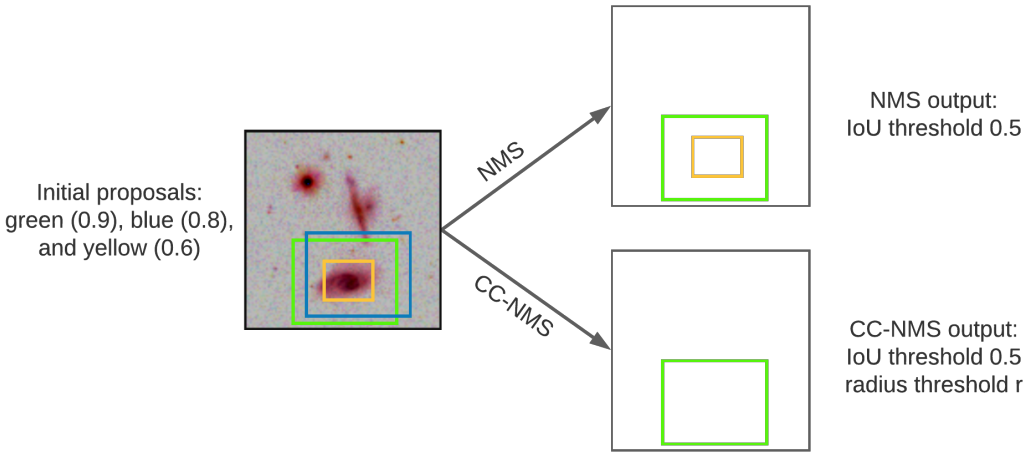


Figure 5.7: CC-NMS vs. NMS: An example of three region proposals with confidence scores.

we propose the CC-NMS module (Connected Component NMS) to determine duplication by comparing the detection centers.

For object detection on natural images, NMS has significantly removed duplicated detection while retaining true-positive detections. Roughly speaking, the NMS strategy leverages IoU scores and confidence scores of detected boxes. The strategy iteratively selects the most confident boxes and removes highly overlapped boxes.

In addition to NMS, the CC-NMS module pays attention also to the center of the detection because astronomical objects are very well centralized. The object's center can be defined as the center mass, the brightest intensity, or the pixel with the highest confidence score in the prediction mask. First, CC-NMS guarantees to detect boxes that NMS has detected. Second, CC-NMS further investigates the centers to find duplication. An ablation study in Sec. 5.6 shows that CC-NMS is suitable to eliminate duplicated detection in astronomical datasets.

Algorithmically, the difference between CC-NMS and NMS is highlighted in Alg. 5 where CC-NMS makes use of detection centers to differentiate object boxes/proposals *(line 8)*. In detail, CC-NMS utilizes three criteria to filter out duplicated proposals: confidence score of boxes, IoU scores between box centers, and Euclidean distance between boxes. Similar to NMS, CC-

Algorithm 5: CC-NMS

Input : A set of initial detected boxes $B = \{b_1, \ldots, b_N\}$,
Input : Corresponding confidence scores $S = \{s_1, \ldots, s_N\}$,
Input : A function to identify center of box $center()$,
Input : IoU threshold $\lambda \in [0, 1]$,
Input : Radius threshold r.
Output: List of filtered boxes F.

$F \leftarrow \{\}$
while $B \neq \{\emptyset\}$ **do**
  `/* Select the most confidence box */`
  $m \leftarrow \arg\max(S)$
  $F \leftarrow F \cup b_m$
  $B \leftarrow B \setminus b_m$
  $S \leftarrow S \setminus s_m$
  `/* Filter out boxes highly overlapped or sharing similar centers with the selected box */`
  foreach $b_i \in B$ **do**
    if $IoU(b_i, b_m) \geq \lambda$ *or* $||center(b_i), center(b_m)|| \leq r$ **then**
      $B \leftarrow B \setminus b_i$
      $S \leftarrow S \setminus s_i$
return F

NMS first ranks all the boxes by confidence scores. Afterward, CC-NMS iteratively selects the most confident box and removes highly overlapped boxes or proximal boxes. The highly overlapped boxes are the boxes that have IoU scores greater than the threshold λ with the currently selected box. A box is considered as a proximal box to the current box if the Euclidean distance between the two centers is less than the threshold r. The procedure continues until all the boxes are visited.

Intuitively, Fig. 5.7 shows a typical example of astronomical object detection where the bright object is usually recognized multiple times at different box scales. Mainly, green and blue boxes are at similar scales, then both NMS and CC-NMS can spot this duplication by checking the intersection using IoU scores. The blue box is eliminated since it has a lower confidence score. Besides, the yellow box represents the same object, but at a much

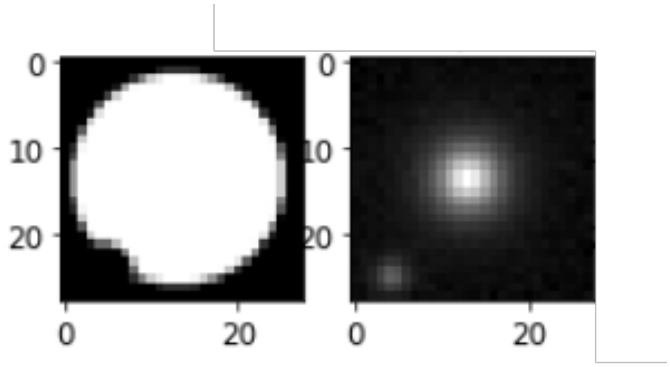


Figure 5.8: A binary ground-truth mask (left) and the corresponding detection box (right).

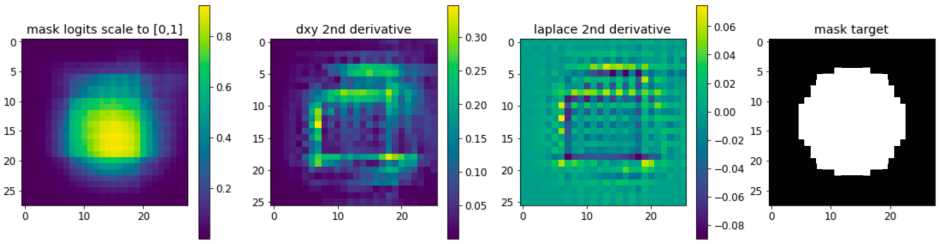
lower scale, then the IoU scores between the yellow and either the blue or the green could never reach the IoU threshold. Hence, NMS could not identify the yellow duplication. On the other hand, CC-NMS also compares the yellow and green centers, which quickly points out they share similar centers. Since the yellow box has a lower confidence score, the green box is kept.

5.3.3 Mask Head Smoothness

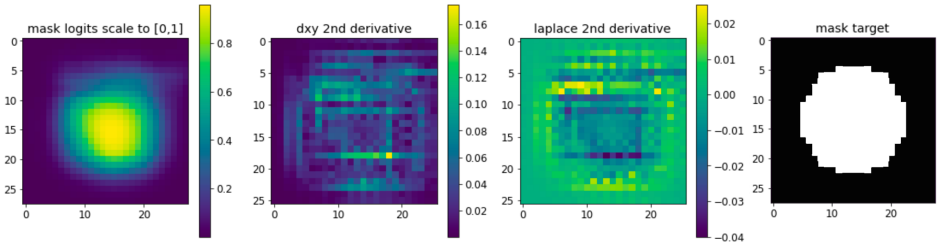
In contrast to natural objects with clear borders, astronomical objects are centralized and expected to be smooth in the extent. Hence, this section proposed to use a gradient regularization term in the mask head to address the smooth segmentation mask.

In the field of machine learning, object segmentation is generally handled by a learning Fully Convolutional Network FCN to preserve a pixel-to-pixel correspondence from input space to segmentation space. Particularly in R-CNN models, the mask head plays the role to map the detection box to the binary ground-truth mask, as shown in Fig. 5.8. The mask head is designed as a sequence of convolutional layers, followed by de-convolutional layers [Dumoulin and Visin, 2016]. The former layers down-sample input images to lower-dimensional features while the latter layers up-sample the features to the segmentation space. Since convolutional and de-convolutional layers naturally retain the input's spatial information, these designs have been used widely for segmentation tasks.

However, *checkerboard artifacts* and *non-smooth masks* remain the two



(a) The model without smoothness regularizer.



(b) The model with smoothness regularizer.

Figure 5.9: Regularizer effect on the mask head of the model: From left to right, we have: predicted mask; the pixel-wise summation of the second derivative of predicted mask with respect to x and y coordinates; the response of predicted mask on a 3×3 Laplacian Kernel as an approximation of the second derivative; and the corresponding ground-truth mask encoded as a binary map.

disadvantages of the mask head architecture for astronomical applications. Both effects can be seen in the left-most mask in Fig. 5.9a. First, the strange checkerboard pattern of artifacts is caused by the de-convolutional layer where the de-convolutional operator has uneven overlap pixels. Second, the non-smooth predicted mask is irregular and unrealistic for astronomical objects as they are expected to have softened segmentation. In fact, the classical mask head produces non-smooth masks because the training set encodes ground-truth masks as binary masks with clear borders. Also, the default mask head's loss function has no attention on the smoothness of the predicted masks.

To generate smooth masks for astronomical objects, our idea is to use the mask gradient to penalize both the non-smooth mask and the checkerboard artifact. The mask loss function L_{mask} only includes the k_{th} mask if the

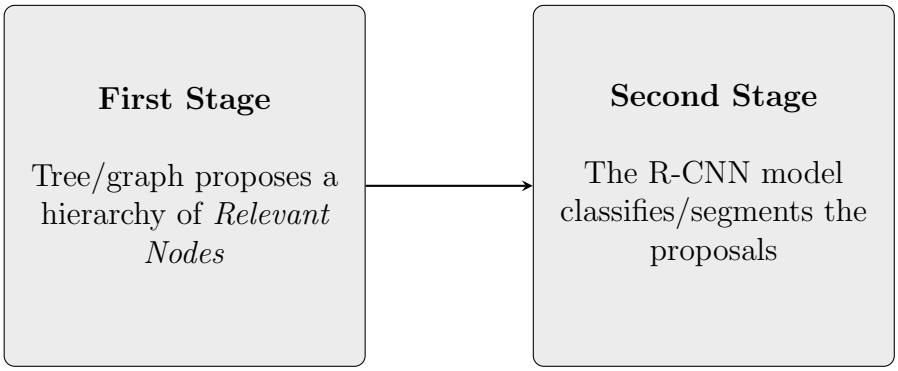


Figure 5.10: The hybrid approach overview.

region is associated with the ground truth class k. Formally,

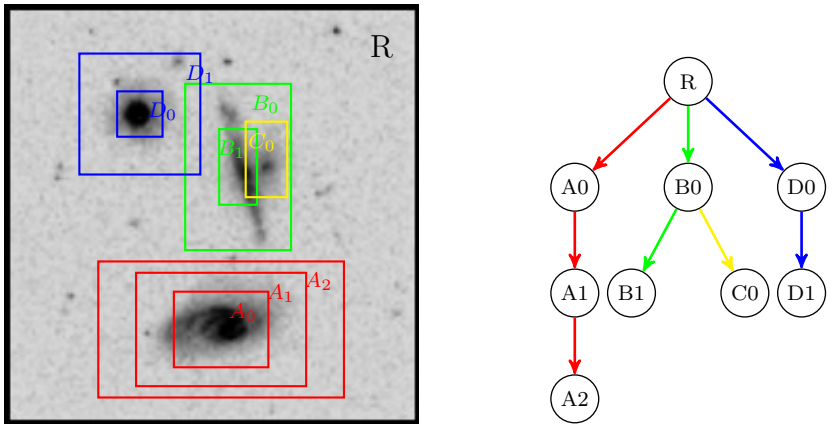
$$\begin{aligned}\mathcal{L}_{\text{mask}}(y,\hat{y}) = &-\frac{1}{m^2}\sum_{1\leq i,j\leq m}\left[y_{ij}\log\hat{y}^k_{ij} + (1-y_{ij})\log(1-\hat{y}^k_{ij})\right] \\ &+\left|\frac{d^2\hat{y}}{dy^2}\right| + \left|\frac{d^2\hat{y}}{dx^2}\right|,\end{aligned} \tag{5.4}$$

where y_{ij} is the label of a $cell_{(i,j)}$ in the true mask for the region of size (m,m); $\hat{y}^k_{ij}$ is the predicted value of the same cell; and $|\frac{d^2\hat{y}}{dy^2}|$, $|\frac{d^2\hat{y}}{dx^2}|$ are the norms of the second derivatives of the mask.

As can be seen in Fig. 5.9, the use of the gradient regularizer helps to produce more realistic and more smooth segmentation compared to the classical mask head. Despite its simplicity, the gradient regularizer visually shows excellent results for astronomical-like objects where segmentation smoothness is essential.

5.4 Hybrid-approach with Tree Proposals

In this section, we propose a hybrid approach using both morphology and ConvNet for astronomical object detection. The hybrid approach can be considered a two-stage detection model. The first stage leverages image hierarchies to select region proposals, while the second stage uses ConvNet-based heads to extract relevant information from the proposals. The motivation of the hybrid approach is presented in Sec. 5.4.1. Then the proposed tree-based proposal module and the complete model are described in Sec. 5.4.2.



(a) Region proposals of an image. (b) Corresponding proposal hierarchy.

Figure 5.11: TPM models region proposals as a hierarchy: Root node represents the whole image; Each box/region proposal is corresponding to a node in the tree; Node parentship shows the inclusion relation between the region proposals in image space.

5.4.1 Motivation of the Hybrid-approach

So far, we have addressed astronomical object detection with two separated directions: CGO - relying on component-graphs in Chapter 3 and the ConvNet-based model in Sec. 5.3.

The proposed CGO framework has the advantage of detecting and organizing objects in a well-structured component-graph, but the latter segmentation stage has difficulty in crowded scenes. Because the morphological hierarchies purely rely on thresholding, then the components are usually under-segmented or over-segmented representations of the objects in the case of interacting objects, see Fig. 1.8.

In contrast, the ConvNet-based model is doing very well at the classification and segmentation stage. However, the first region proposal stage always faces multiple proposals of a single object, i.e., false positives. In that case, NMS and CC-NMS are the popular solutions to get rid of the false-positive proposals based on IoU scores and detection confidences.

Based on these observations, we think about a hybrid model where both morphology and ConvNet take part in the stage that they have the advantage. Figure 5.10 presents the abstraction of the hybrid idea in a two-stage architecture. Concretely, morphology is reserved for the first stage of the re-

gion proposal, while ConvNet responds for the second stage of classification and segmentation. The prediction heads in the second stage are similar to the second stage in the proposed R-CNN-based model. On the other hand, the first stage is not that straightforward, and it requires integrating the tree/graph structures into the general model. The integration will be handled by a Tree-based Proposal Module (TPM), described in the following Sec. 5.4.2.

Compared to the attention mechanism in RPN, the TPM module falls back to classical rule-based approaches to select proposals. However, TPM not only selects proposals but also retains the relationship between proposals as hierarchies. These hierarchies will be used to eliminate multiple detections in the later stage. Besides, it is possible to train an RPN-like network on the well-structured TPM proposals.

Intuitively, the TPM module attempts to achieve reasonably accurate proposals with image hierarchies. To be clear, we consider two types of false positives: *Type 1* - wrong detections and *Type 2* - multiple detections on a single object. TPM tolerates the *Type 1* false-negative proposals as these proposals' objectness will be assessed efficiently in the second stage. The key point of TPM is to retain the proposal hierarchy because a branch of the hierarchy reflects the multiple representations of a single object. This hierarchical information helps to get rid of the remaining *Type 2* false-positive proposals. As can be seen in the example in Fig. 5.11, region proposals of the input image are organized as a tree where we can suspect multiple boxes of the same object falling to the same branch of the tree. Roughly speaking, TPM in the first stage tends to filter out *Type 2* false-positive proposals, while the second stage is better at removing *Type 1* false-negative proposals.

5.4.2 Proposed Tree-based Proposal Module

This section introduces a Tree-based Proposal Module (TPM) to integrate morphology into the first stage of the hybrid model to select potential regions.

We propose to firstly filter morphological representations of the input image with statistical hypothesis tests and filtering strategies introduced in [Nguyen et al., 2020, 2021; Teeninga et al., 2016] but with lower significance levels or confidence thresholds. The low confidence thresholds aim at covering all objects without taken care of wrong false-positive proposals (i.e., *Type-1* false positives). From the initially selected proposals, we focus on two proposal attributes: area and center. The area is roughly estimated by the

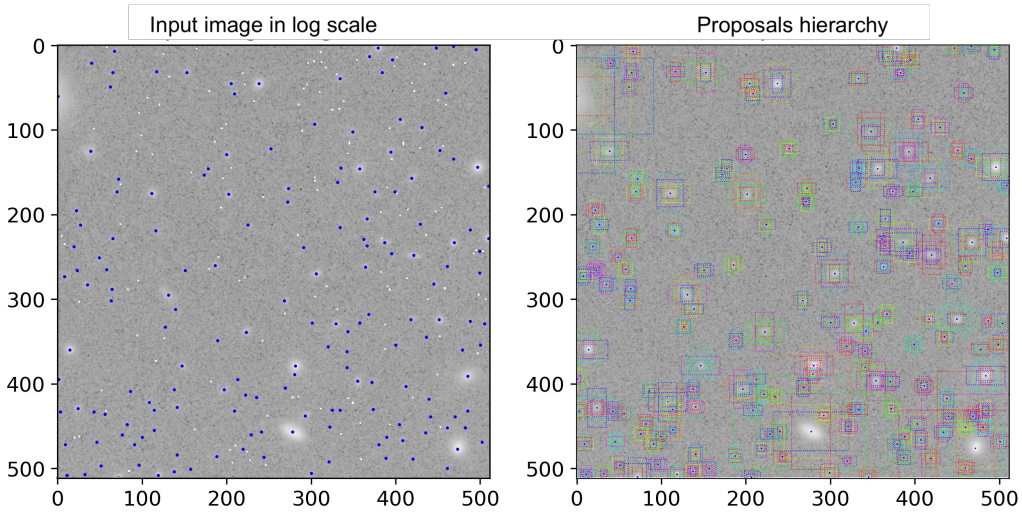


Figure 5.12: TPM: (left) Centers of initially selected proposals and (right) multi-scale anchors of the selected proposals.

number of component pixels, while the center can be obtained by following the main branch of the proposal, see Fig. 5.12.

Second, we propose to build a proposal hierarchy based on the proposal areas and centers, see Fig. 5.11. We generate multi-scale boxes/anchors for each initially selected proposal by defining box scales and box aspect ratios. To note that anchor parameters are defined relatively to area groups (small, medium, or large). Even though the area is not accurately estimated via the connected components, it helps to avoid strange anchors like huge anchors for tiny proposals or vice versa. Practically for each group by area, we define three scales and three aspect ratios yielding nine anchors associate to the selected proposal, as can be seen in Fig. 5.12. The multi-scale anchors can be thought of as a little pyramid associated to each selected proposal.

All in all, the TPM outputs consist of the anchor pyramids and the hierarchy of selected proposals. On the one hand, the generated anchors can be used to feed the second stage for training. On the other hand, the pyramids and hierarchy will help remove the *Type 1* false positives of multiple detections of a single object. Precisely, the second stage assigns objectness scores to the proposed anchors. Since the anchor pyramids are retained, we can filter out low-confident anchors in each pyramid, i.e., select at most one representative anchor for each pyramid.

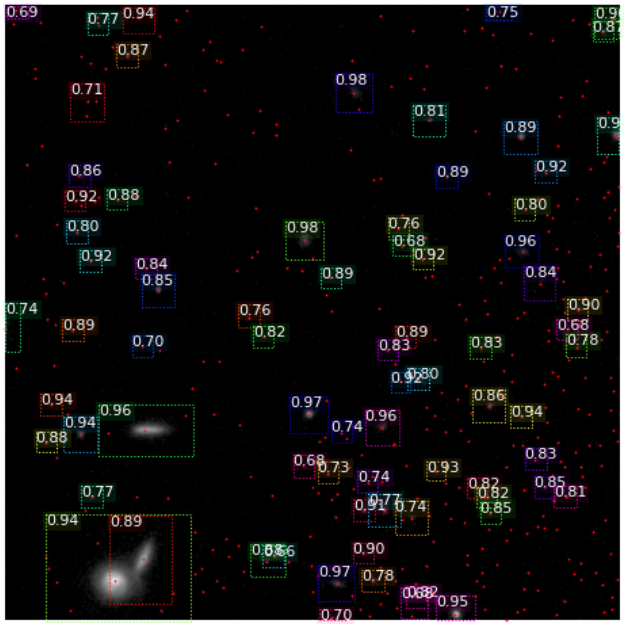
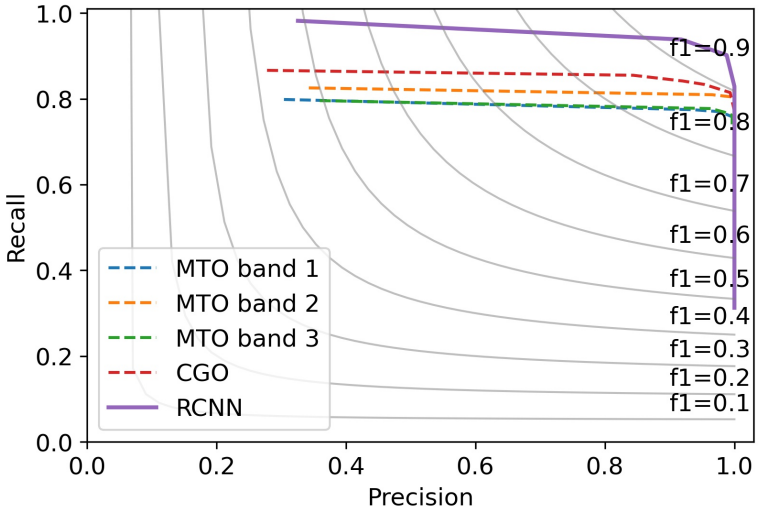


Figure 5.13: The hybrid approach: colorized boxes present final detected objects with confidence scores; red dots are centers of initially selected proposals in the TPM module.

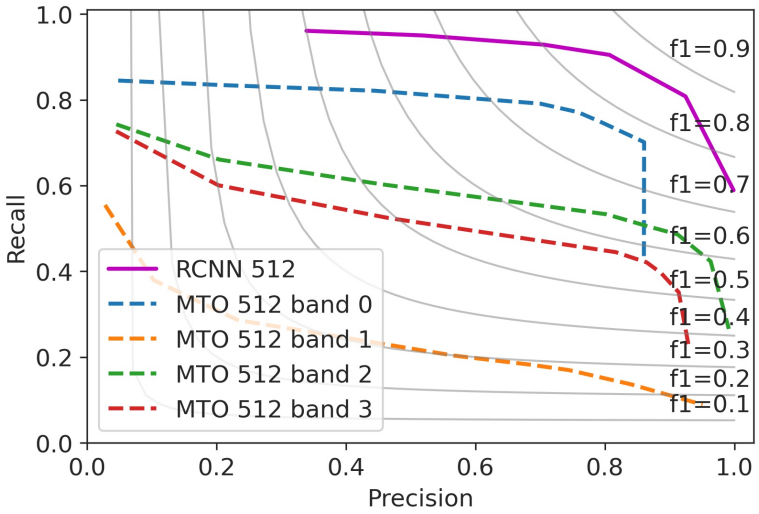
Compared to RPN constructing multi-scale anchors on the image grid, the TPM module constructs multi-scale anchors of the initially selected proposals. Initial experiments have shown comparable results between the hybrid model using TPM and the proposed R-CNN-based model using RPN, see Sec. 5.6.5. Visually, as can be seen from the example in Fig. 5.13, objects in interacting regions are well identified and organized.

5.5 Experiments

We perform thorough comparisons between our R-CNN-based proposal and the state of the art [Teeninga et al., 2016; Wilkinson et al., 2019] on both simulated and real datasets. We first define the evaluation metric in Sec. 5.5.1, then discuss the main detection results in Sec. 5.5.2 and Sec. 5.5.3.



(a) RCNN vs. MTO, CGO on the Simulation.



(b) RCNN vs. MTO on KiDS dataset: size 512×512 images.

Figure 5.14: Experimental results.

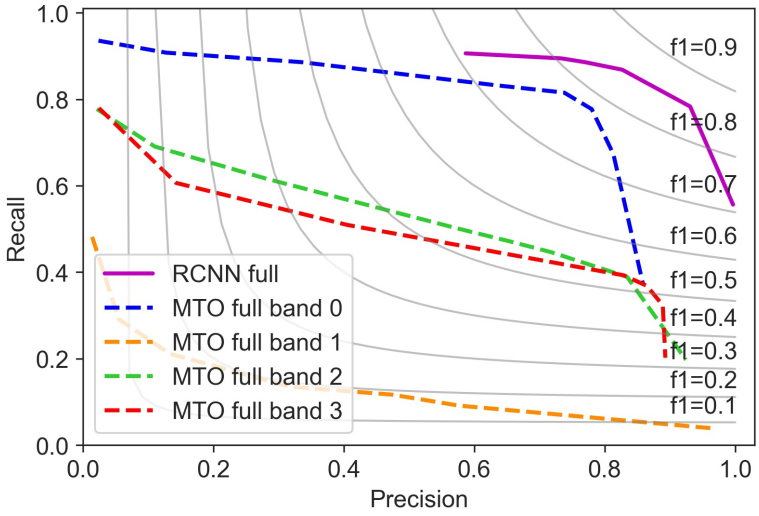
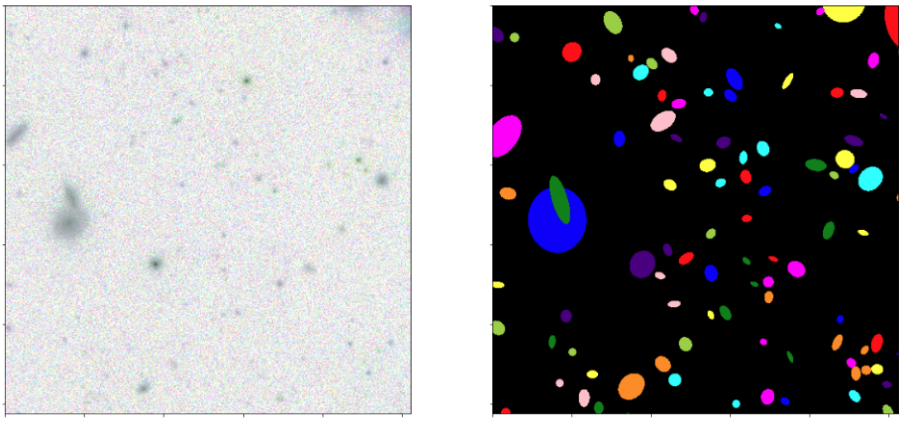
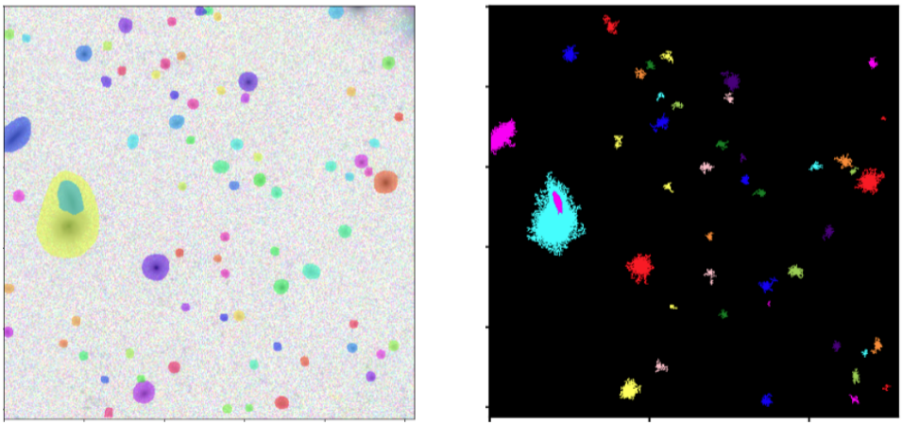


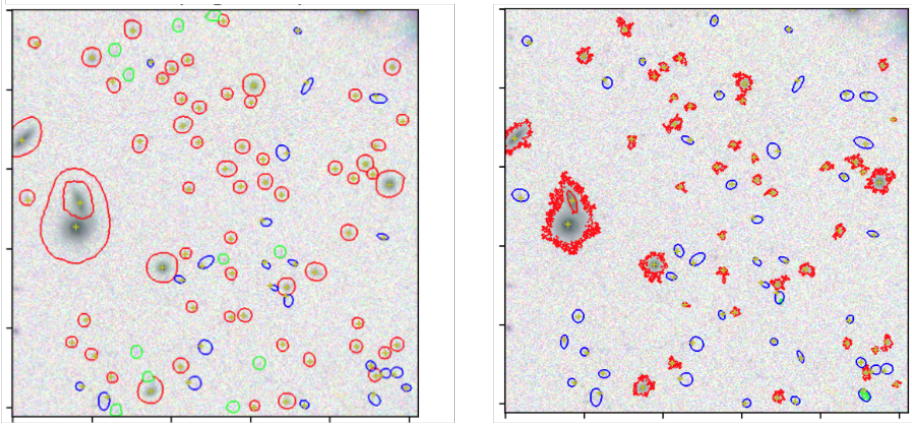
Figure 5.15: RCNN vs. MTO on KiDS dataset: full size images.



(a) Input: (left) A real KiDS image and (right) annotated ground-truth objects.



(b) Output segmentation: (left) R-CNN and (right) MTObject.



(c) Matching results with the ground-truth: (left) RCNN and (right) MTObject.

Figure 5.16: Visual assessment example: R-CNN with $confidence > 0.5$ and MTObject with $\alpha = 10^{-6}$ $move\_factor = 0.5$; Red, green, and blue polygons present true positive, false positive, and false negative detections.

5.5.1 Evaluation metric

We use precision, recall, and F1-score, the same as in [Haigh et al., 2020]. The evaluation matches at most one detected object in the detection map to each target object in the ground-truth map. Each target object in the ground-truth map is represented by its brightest pixel. Hence each representative pixel is included in at most one object in the detection map. Suppose a detected object contains several representative pixels of different target objects. In that case, the detected object is associated to the target object with the brightest representative pixel, more detail in Chapter 3 Section 3.5.3.

5.5.2 Experiment on simulated dataset

We compare the proposed model versus MTObject [Teeninga et al., 2016] and CGO [Nguyen et al., 2020, 2021] on the FDS Simulation. Since the signal close to the image's border is less reliable, we skip objects whose center is lying within 100 pixels from the borders. Precision and recall curves are presented in Fig. 5.14a.

Given the simplicity of the simulated images, all methods achieve favourable recall with almost any choice of model hyper-parameters. However, our proposed model significantly improves MTObject and CGO at all metrics (precision, recall, and F1 score) in the FDS Simulation.

5.5.3 Experiment on real dataset

In this test, we focus on comparing the proposed RCNN model versus MTObject on the real KiDS images. Because of computational complexity limitation, CGO is currently inefficient to perform the analysis on full real images. The RCNN model has been trained on four-band KiDS images of size 512×512 while MTObject requires no training. For evaluation, the RCNN model is tested on a fixed-size 512×512 image dataset. RCNN evaluation on variable size images is possible with a slight drop of performance, more detail in sec. 5.19. On the other hand, MTObject is tested on the same fixed size images and full-size images, results are reported in Fig. 5.14b and Fig. 5.15.

It is clear that the RCNN model outperforms both MTObject on fixed and variable size real images. Given the more complicated real image structures, the gap between RCNN and MTObject results expands significantly

compared to the experiment on simulated images. MTObject results also indicate the variation between results on separated bands.

For visual assessment, an example result on real image is presented in Fig. 5.16. Apart from the detection precision and recall trade-off, which is fully depicted in the curves, we can see clearly that the Object Masks/Segmentation is the main drawback of the morphological approaches. Particularly, masks of small objects and superimposed objects are generally under-segmented.

5.6 Ablation Studies

We run a number of comprehensive ablation studies to analyze the proposed R-CNN based model, including:

1. **Multi-band input images**: we inspect the effects of input size and the number of the bands (or channels) of input images on the model performance.

2. **Variable-size input images**: we evaluate the model with variable-size input images.

3. **Normalization layer**: to compare the normalization layer and hand-craft normalization pre-processing.

4. **CC-NMS module**: to compare the use of the default NMS and CC-NMS.

5. **Tree-based proposal module (TPM)**: to study the differences between TPM and RPN.

5.6.1 Multi-band Input Images

First, this ablation test compares our proposed R-CNN model training and testing on different datasets with different numbers of input image bands. Generally, information gain from increasing the number of bands usually leads to better model performance. Our proposed model CGO (in Chapter 3 and [Nguyen et al., 2020, 2021]) outperformed the state-of-the-art MTObject [Teeninga et al., 2016] for the same reason.

In detail, the same R-CNN-based model is trained and evaluated on two separated datasets: three-band KiDS images and four-band KiDS images.

Precision and recall of the two models in Fig. 5.17 confirm the expected effect of the additional band in the four-band dataset. In the test, the additional fourth band indeed has the lowest signal-to-noise ratio, i.e., the lowest image quality compared to other bands. The fourth band stands alone is not useful, but the combination of the fourth band and the others boosts the detection performance. It clearly shows that the multi-band processing plays an essential role in the detection model.

Second, we experiment with the same RCNN model on fixed-size input datasets: 512×512, 768×768, and 1024×1024 four-band KiDS images. The KiDS image dataset originally contains variable-size multi-band images, and the fixed-size datasets are generated by augmenting (crop, rotate, flip) the original ones. Evaluation results in Fig. 5.18 show preferences to the smaller size datasets (512×512 and 768×768). This can be explained by the limitation of the small KiDS dataset (it is small in terms of both image size and number of images). On the one hand, the large size (1024×1024) unexpectedly decrease the number of KiDS images that satisfied the size criterion. On the other hand, the large size also makes some augmentations more difficult, such as rotated crop usually going out of the original image, leading to unnatural region filling by zero value. Practically, our proposed RCNN-based model currently works better on small fixed-size input datasets.

5.6.2 Variable-size Input Images

This ablation targets to see the RCNN model performance on variable size input images. We have chosen the best model trained on a fixed size dataset of four-band 512×512 KiDS images (as shown in Fig. 5.18). Afterward, the model is evaluated on variable size four-band input image sizes: 512×512, ..., 1152×1152.

R-CNN-based models are input size-independent, i.e., these models accept variable size input images at both training and inference. Precisely, three main components in the RCNN model are all input size-independent: First, the feature extractor backbone is purely convolutional, then variable-size inputs just produce variable-size features; Second, RPN proposes a fixed number of regions no matter what size of input features; Lastly, prediction heads received fixed sized region features, then the last component is entirely not related to the input image size.

Even though the RCNN model can be trained with variable size input, it is recommended to train with fixed-size images [Girshick, 2015]. For infer-

ence, we practically observe that detection performance drops when testing large images with the model trained on fixed-size images. For these cases, objects in the large testing image are likely to outnumber objects in the fixed-size training image while RPN fixes the number of proposals. Based on that observation, we propose to adapt the number of RPN proposals proportional to the input image area at inference. The adaptation helps the model perform reasonably well if testing image sizes are significantly different from the trained image size.

The variable size evaluation of the model trained on fixed-size images is shown in Fig. 5.19. As we can see, the model performs best on image 512×512 because it has been trained at the same size images. However, it is worth noting that the performance of variable size inputs only drops slightly. Interestingly, this ablation shows that the RCNN based model can be trained on fixed-size images, but can inference on variable-size images without significant performance drops.

5.6.3 Normalization Layer

We have experimented the RCNN-based model with and without the normalization layer. Apart from the normalization, the two models are the same, being trained and evaluated on 512×512 four-band KiDS images. The results are shown in Fig. 5.20. We found out the normalization layers are fragile to train from the beginning. It could be a consequence of having clipping and gamma operators at very early layers. Practically, we fix the normalization parameters at initial training, then gradually release these parameters at later training stages.

Despite the simplicity of the normalization layer, Fig. 5.20 depicts that the model with the normalization layer outperforms the other. On the other hand, the normalization layer also gives a quick and good visualization for astronomical images instead of manual processing.

5.6.4 CC-NMS module

This ablation aims at showing the difference between NMS and CC-NMS modules. The two models are trained and evaluated on 512×512 three-band simulated images, results are reported in Fig. 5.21. By adding the center criterion, the model with CC-NMS can filter out false positives to get reasonable precision (≥ 0.7) given any choice of model hyperparameters. In

return, CC-NMS inevitably sacrifices some true-positive detections. As we can see, both models have favourable F1-scores, but there is a trade-off: the model with CC-NMS achieves better precisions while the model with NMS has a better recall in some cases.

5.6.5 Tree-based Proposal Module (TPM)

We experimented with two models: the default RCNN with Region Proposal Network (RPN) and the hybrid with Tree-based Proposal Module (TPM). As can be seen from Fig. 5.22, the hybrid model has comparable results to the RCNN approach while both models undoubtedly surpass the morphological method baseline.

Even though the hybrid evaluation result has not outperformed the RCNN model with RPN yet, we believe the hybrid is the interesting direction to investigate for two reasons. First, TPM interestingly holds the hierarchical relation of proposed regions/objects; Second, TPM helps to get rid of the NMS and CC-NMS modules to remove multiple detections.

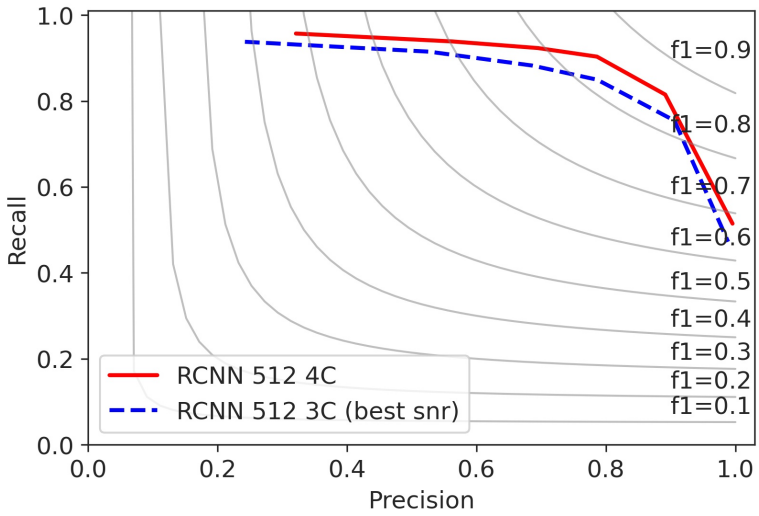


Figure 5.17: Ablation study on number of channel of input images

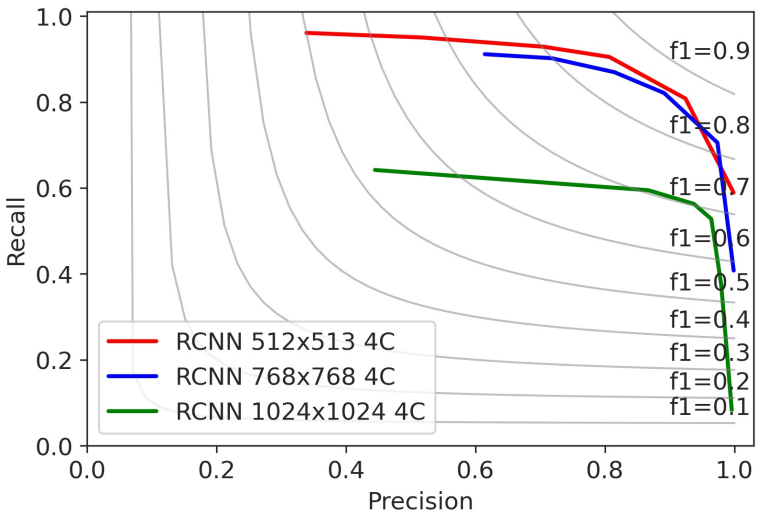


Figure 5.18: Ablation study on size of input images

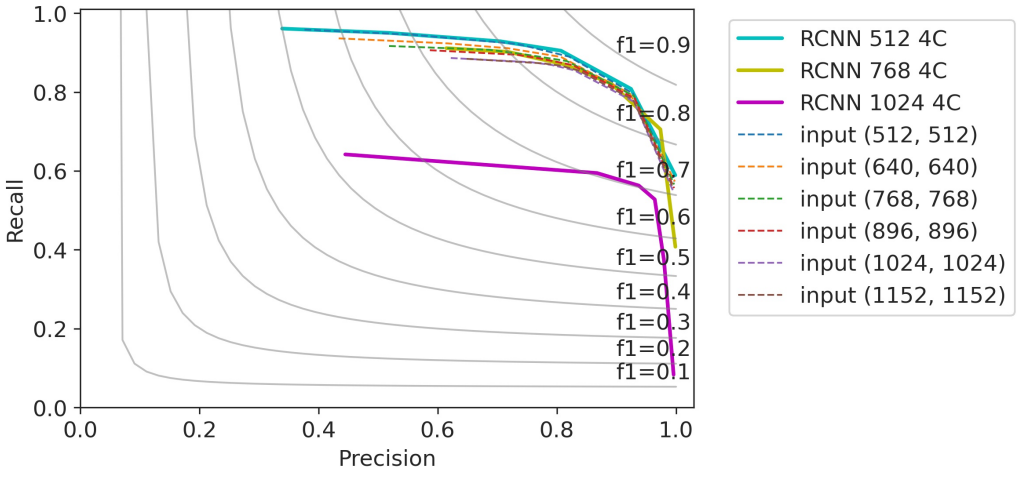


Figure 5.19: Ablation study on variable size input images effects.

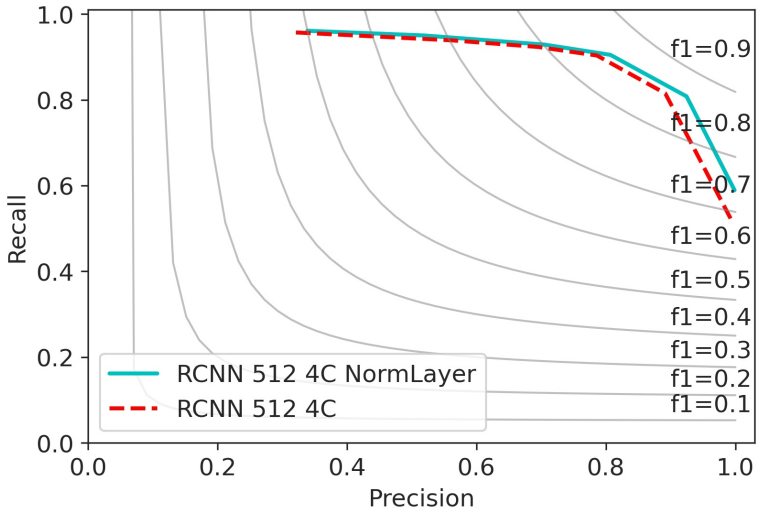


Figure 5.20: Ablation study of the normalization layer.

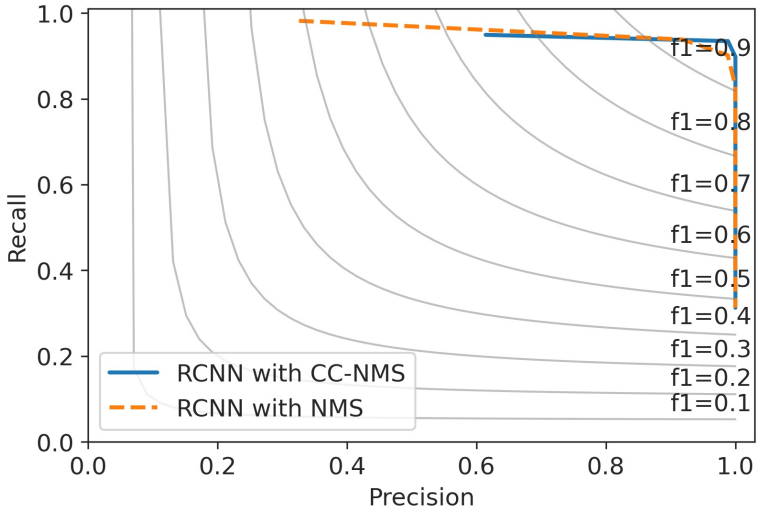


Figure 5.21: Ablation study on CC-NMS and NMS.

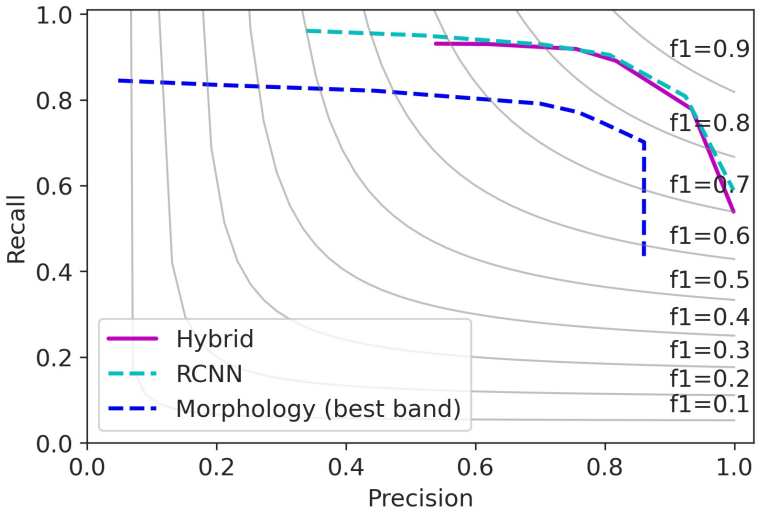


Figure 5.22: RCNN with Tree-based Proposal Module.

5.7 Conclusion and Perspectives

In conclusion, we introduced a real dataset of multi-band astronomical objects with annotated ground-truth. The dataset and the customized annotation tools are publicly available [†]. Given the dataset, we have proposed an RCNN-based model tailored to detect astronomical objects. There are three novel components in the proposed model, including a normalization layer, a CC-NMS module, and a smoothness regularizer. Experiments show that our proposed model significantly outperforms the state of the art on both synthetic and real datasets. Besides, we have investigated a hybrid approach that uses both morphological-based and RNN-based models to adapt to astronomical contexts. Initial experiments show comparable results to the proposed RCNN approach. However, the hybrid approach's hierarchy information is the potential direction to get rid of the NMS/CC-NMS module for multiple detection removal tasks.

[†] https://github.com/hetpin/sky_imview

Bibliography

Bertin, E. and Arnouts, S. (1996). Sextractor: Software for source extraction. *Astronomy and Astrophysics Supplement Series*, 117:393–404.

Burke, C. J., Aleo, P. D., Chen, Y.-C., Liu, X., Peterson, J. R., Sembroski, G. H., and Lin, J. Y.-Y. (2019). Deblending and classifying astronomical sources with mask r-cnn deep learning. *Monthly Notices of the Royal Astronomical Society*, 490(3):3952–3965.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee.

Dumoulin, V. and Visin, F. (2016). A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*.

Dutta, A., Gupta, A., and Zissermann, A. (2016). Vgg image annotator (via). *URL: http://www. robots. ox. ac. uk/˜ vgg/software/via*.

Dutta, A. and Zisserman, A. (2019). The via annotation software for images, audio and video. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2276–2279.

Farias, H., Ortiz, D., Damke, G., Arancibia, M. J., and Solar, M. (2020). Mask galaxy: Morphological segmentation of galaxies. *Astronomy and Computing*, page 100420.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2009). Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). 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.

Haigh, C., Chamba, N., Venhola, A., Peletier, R., Doorenbos, L., Watkins, M., and Wilkinson, M. (2020). Optimising and comparing source extraction tools using objective segmentation quality criteria. *Astronomy and Astrophysics*.

Hausen, R. and Robertson, B. E. (2020). Morpheus: A deep learning framework for the pixel-level analysis of astronomical image data. *The Astrophysical Journal Supplement Series*, 248(1):20.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7310–7311.

Hubble (2000). The hubble legacy archive. https://hla.stsci.edu/.

Koekemoer, A. M., Faber, S., Ferguson, H. C., Grogin, N. A., Kocevski, D. D., Koo, D. C., Lai, K., Lotz, J. M., Lucas, R. A., McGrath, E. J., et al. (2011). Candels: the cosmic assembly near-infrared deep extragalactic legacy survey—the hubble space telescope observations, imaging data products, and mosaics. *The Astrophysical Journal Supplement Series*, 197(2):36.

Kuijken, K., Heymans, C., Dvornik, A., Hildebrandt, H., de Jong, J., Wright, A., Erben, T., Bilicki, M., Giblin, B., Shan, H.-Y., et al. (2019). The fourth data release of the kilo-degree survey: ugri imaging and nine-band optical-ir photometry over 1000 square degrees. *Astronomy & Astrophysics*, 625:A2.

Nguyen, T., Chierchia, G., Razim, O., Peletier, R., Najman, L., Talbot, H., and Perret, B. (2021). Object detection with component-graphs in

multi-band images: Application to source detection in astronomical images. *IEEE Access*, pages 156482–15649.

Nguyen, T. X., Chierchia, G., Najman, L., Venhola, A., Haigh, C., Peletier, R., Wilkinson, M. H. F., Talbot, H., and Perret, B. (2020). Cgo: Multi-band astronomical source detection with component-graphs. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 16–20. IEEE.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

Teeninga, P., Moschini, U., Trager, S. C., and Wilkinson, M. H. (2016). Statistical attribute filtering to detect faint extended astronomical sources. *MMTA*, 1.

Uijlings, J. R., Van De Sande, K. E., Gevers, T., and Smeulders, A. W. (2013). Selective search for object recognition. *International journal of computer vision*, 104(2):154–171.

Venhola, A. (2019). *Evolution of dwarf galaxies in the Fornax cluster*. PhD thesis, University of Groningen.

Venhola, A., Peletier, R., and et al. (2018). The fornax deep survey with the vst-iv. a size and magnitude limited catalog of dwarf galaxies in the area of the fornax cluster. *Astronomy & Astrophysics*, 620:A165.

Wilkinson, M. H. F., Haigh, C., Gazagnes, S., Teeninga, P., Chamba, N., Nguyen, T. X., Talbot, H., Najman, L., Perret, B., Chierchia, G., Venhola, A., and Peletier, R. (2019). Sourcerer: A robust, multi-scale source extraction tool suitable for faint and diffuse objects. In *IAU 355 Symposium*.

Zitnick, C. L. and Dollár, P. (2014). Edge boxes: Locating object proposals from edges. In *European conference on computer vision*, pages 391–405. Springer.

CONCLUSIONS AND PERSPECTIVES

This thesis aims at developing multi-band object detection algorithms with applications to astronomical images. We have proposed three models based on mathematical morphology and convolutional neural networks. Comprehensive analyses on both simulated and real datasets illustrated the efficiency and the performance of the proposed approaches compared to the state of the art. To conclude this manuscript, we review our contributions and discuss future work perspectives.

Conclusion

The first part of the thesis has addressed the object detection problem with morphological approaches. We have introduced CGO - a novel morphological framework for object detection in multi-band images relying on component-graphs and statistical hypothesis tests. The framework has been applied to detect multi-band astronomical sources on both simulation and real astronomical images. Experiments have demonstrated a significant improvement in detecting objects on both multi-band simulated and real astronomical images. The main contributions of the morphological approach include:

- Proposed CGO - a novel multi-band object detection framework relying on component-graphs and application to astronomical source detection.
- Addressed that the component-graph is better at capturing image structures comparing to classical component-trees.
- Introduced two filtering algorithms to detect duplicated and partial nodes in the component-graphs.
- Improved object detection results on simulated and real multi-band astronomical images.

The second part turned our attention to the class of ConvNet approaches and hybrid approaches. Experiments show that our proposed model significantly outperforms the state of the art. Briefly, the second part has brought four main contributions:

- Introduced a real dataset of multi-band astronomical objects with annotated ground-truth. The idea to use higher quality images to annotate lower quality images semi-automatically.
- Proposed an RCNN-based model tailoring object detection on astronomical images. The novelties of the proposed model consist of: a trainable normalization layer that can be trained end-to-end with the whole model; CC-NMS module is designed to replace the default NMS at removing multiple detections of a single object; and a smoothness regularizer for the segmentation head in the model.
- The proposed R-CNN-based model outperformed the state of the art on both fixed-size and variable-size real images.

- Proposed a hybrid approach using both morphological trees and R-CNN models for object detection. It utilizes a morphological-tree to detect potential regions in the first stage, then it uses convolutional heads to predict relevant information such as labels and segmentation masks. Initial experiments showed comparable but potential results to the proposed RCNN approach. The hierarchy information in the hybrid approach is the potential direction to get rid of the NMS/CC-NMS module for multiple detection removal tasks.

Perspectives

In the first part of the manuscript, the proposed morphological approach - CGO - opens up some interesting perspectives for the component-graph.

- Component-graph Filtering Algorithms: Given the proven richness of the component-graph at handling multi-band data, the acyclic graph structures remain challenging for filtering algorithms. We think that *directed connected operators* on directed acyclic graphs [Perret et al., 2014] and *shape space filtering* [Xu et al., 2015; Grossiord et al., 2019] on the component-graph could be interesting to investigate.

- Component-graph Construction Algorithms: Component-graph construction algorithms are in the early development stage as the component-graphs has been introduced not so long ago compared to the classical component-trees [Berger and et al., 2007; Passat et al., 2019]. The directed acyclic structure of the component-graph is the main difficulty preventing the use of many optimization techniques used in component-trees construction algorithms. While a union-find alike strategy is impossible in the graph, we expect paralleled constructions can help. In fact, parallelism has significant speed up building the component-trees by merging multiple sub-trees of image partitions [Wilkinson et al., 2008; Ouzounis and Wilkinson, 2007]. We have developed a version of CGO building the component-graph on partitions of the image, but merging sub-graphs remain challenging.

- Component-graph Attributes: Another direction is to learn the component attributes rather than using predefined ones. New color-based

attributes are also promising to explore since the component-graphs handle multi-band information simultaneously.

The second part of the manuscript shows potential R-CNN-based models' performances and sheds light on hybrid approaches using both morphology and ConvNet. Several interesting directions are open to being discussed:

- Normalization Layer: The normalization layer has demonstrated the benefits of improving detection accuracy and producing an intermediate visualization for astronomical images. The normalization layer's design is highly extendable to other kinds of images, such as applications to medical images would be interesting. Also, we have planned further to develop the normalization layer as a stand-alone visualization module.

- Focal Loss: The weighted loss idea [Lin et al., 2017] from the one-stage object detection models could be interesting to apply in the two-stage R-CNN loss function.

- CC-NMS: The CC-NMS module could initially model boxes as directed graphs where each node represents a box with attributes. Instead of interactively visiting a set of unstructured boxes, we can efficiently filter nodes on the graphs. In addition to attribute filtering, the box inclusions can be easily spotted as branches in the directed graphs.

- Soft-mask Training: As the astronomical object border is soft, we are working toward training the proposed models with a dataset of soft ground-truth masks. Given the current annotated binary masks, the soft ground-truth masks can be obtained by merely convolving the annotated masks with Gaussian kernels or weighting the annotated masks with normalized pixels intensity. We think combining the soft-mask training and the proposed smoothness regularizer can produce more realistic segmentation masks.

- Multi-band Segmentation Mask: Astronomical objects might shine differently on different bands, so it is useful to generate multi-band masks for each object. Naturally, the ConvNet architectures are able to generate the multi-band output by adding multiple prediction heads as expected. The more difficult part is gathering multi-band segmentation ground-truth masks.

- Trainable Tree-based Proposal Module: Despite showing potential benefits comparing to the attention mechanism RPN, the TPM module falls back to classical rule-based approaches to select proposals. It would be interesting to make the TPM module trainable. In other words, we can learn an RPN-like network on the well-structured TPM proposals.

- Objects as points: The object center is vital in the context of astronomical object detection. In addition to predicting the object box, let the model regresses the object center is possible [Zhou et al., 2020]. Given the information of object center, it could help to differentiate duplicated detection boxes.

- Transfer learning: The ConvNet-based models require training while morphological models can be used directly. However, transfer learning [Torrey and Shavlik, 2010] could help to avoid training the ConvNet-based models again from scratch. Given the model pre-trained on the KiDS-HST dataset, we will need much less effort and fewer data to transfer it to a new dataset compared to the initial training.

Bibliography

Berger, C. and et al. (2007). Effective component tree computation with application to pattern recognition in astronomical imaging. In *IEEE ICIP*, volume 4, pages IV–41.

Grossiord, E., Naegel, B., Talbot, H., Najman, L., and Passat, N. (2019). Shape-based analysis on component-graphs for multivalued image processing. *MMTA*, 3:45–70.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollar, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Ouzounis, G. K. and Wilkinson, M. H. (2007). A parallel implementation of the dual-input max-tree algorithm for attribute filtering. In *ISMM (1)*, pages 449–460.

Passat, N., Naegel, B., and Kurtz, C. (2019). Component-graph construction. *JMIV*, 61:798–823.

Perret, B., Cousty, J., Tankyevych, O., Talbot, H., and Passat, N. (2014). Directed connected operators: Asymmetric hierarchies for image filtering and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 37(6):1162–1176.

Torrey, L. and Shavlik, J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global.

Wilkinson, M. H., Gao, H., Hesselink, W. H., Jonker, J.-E., and Meijster, A. (2008). Concurrent computation of attribute filters on shared memory parallel machines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1800–1813.

Xu, Y., Géraud, T., and Najman, L. (2015). Connected filtering on tree-based shape-spaces. *IEEE TPAMI*, 38:1126–1140.

Zhou, X., Koltun, V., and Krähenbühl, P. (2020). Tracking objects as points. In *European Conference on Computer Vision*, pages 474–490. Springer.