Small target detection using deep learning

Alina Ciocarlan

université PARIS-SACLAY

Small target detection using deep learning

Détection de cibles de petite taille par deep learning

Thèse de doctorat de l'Université Paris-Saclay

École doctorale n° 580, Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat: Sciences du Traitement du signal et des images
Graduate School: Informatique et Sciences du Numérique
Référent: Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche **Optique et Techniques Associées** (Université Paris-Saclay, ONERA),
sous la direction de **Sylvie LE HEGARAT-MASCLE**, Professeure des Universités,
le co-encadrement de **Sidonie LEFEBVRE**, ingénieure de recherche,
et de **Clara BARBANSON**, ingénieure

Thèse soutenue à Paris-Saclay, le 10 décembre 2024, par

Alina CIOCARLAN

Composition du jury

Membres du jury avec voix délibérative

Yann GOUSSEAU Professeur Télécom Paris/LTCI	Président
Sébastien DESTERCKE Directeur de recherche CNRS - UTC/Heudiasyc	Rapporteur & Examinateur
Ronan FABLET Professeur IMT Atlantique/Lab-STICC	Rapporteur & Examinateur
Sébastien LEFEVRE Professeur Université de Bretagne Sud/IRISA	Rapporteur & Examinateur
Céline HUDELOT Professeure CentraleSupélec/MICS	Examinatrice

THESE DE DOCTORAT

NNT : 2024UPASG102

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Détection de cibles de petite taille par deep learning
Mots clés: Paradigme *a contrario*, apprentissage auto-supervisé, YOLO, segmentation sémantique

Résumé: La détection de petits objets dans les images infrarouges (IR) est une tâche complexe mais cruciale en défense, surtout lorsqu'il s'agit de distinguer ces cibles d'un fond texturé. Les méthodes de détection d'objets classiques peinent à trouver un équilibre entre un taux de détection élevé et un faible taux de fausses alarmes. Bien que certaines approches aient amélioré les réponses des cartes de caractéristiques pour les petits objets, elles restent tout de même sensibles aux fausses alarmes induites par les éléments du fond. Pour résoudre ce problème, la première partie de cette thèse introduit un critère de décision *a contrario* dans l'entraînement des réseaux de neurones. Cette méthode statistique améliore les réponses des cartes de caractéristiques tout en contrôlant le nombre de fausses alarmes (NFA) et peut être intégrée dans n'importe quel réseau de segmentation sémantique. Le module NFA améliore la détection des petits objets et renforce la robustesse dans des contextes d'apprentissage frugal en données. Cependant, les réseaux de segmentation peuvent entraîner une fragmentation des objets, causant ainsi des fausses alarmes et faussant les métriques de comptage. Pour atténuer cela, le critère *a contrario* a été intégré dans la tête de détection d'un YOLO. La deuxième partie de la thèse aborde les défis posés par le manque de données annotées grâce à l'apprentissage auto-supervisé (SSL). Nous avons réalisé une étude des catégories de SSL existantes, en mettant l'accent sur les méthodes adaptées à la détection de petits objets. Nous avons ensuite évalué plusieurs stratégies SSL sur différents jeux de données, y compris les datasets de détection de petites cibles en IR. Cette étude nous permet de proposer une feuille de route pour aider à la sélection d'une stratégie de SSL adéquate selon plusieurs paramètres. Enfin, la combinaison du SSL et du paradigme *a contrario* a donné des résultats impressionnants sur la détection de petites cibles en IR.

Title: Small target detection using deep learning
Keywords: *a contrario* paradigm, self-supervised learning, YOLO, semantic segmentation

Abstract: Detecting small objects in infrared images is a challenging yet critical task in defense, especially when it comes to differentiating these targets from a noisy or textured background. Conventional object detection methods have difficulties in finding the balance between high detection rate and low false alarm rate. While some existing approaches have improved feature map responses for small objects, they frequently fail to manage false alarms caused by background elements. To address this, we introduce an *a contrario* decision criterion into neural network training. This statistical test enhances feature map responses while controlling the number of false alarms (NFA) and can be integrated into any semantic segmentation network. The NFA module improves infrared small target detection (IRSTD) and increases robustness in few-shot settings. However, segmentation networks can lead to object fragmentation, causing false alarms and distorting counting metrics. To mitigate this, the *a contrario* criterion has been integrated into the YOLO detection head. The second part of the thesis focuses on overcoming the challenges of limited annotated samples through self-supervised learning (SSL). To this end, we conduct a survey on SSL strategies for image representation learning, with an emphasis on methods adapted for small object detection. We then benchmark several SSL strategies across different datasets, including IRSTD datasets. This study allows us to provide a roadmap to guide future practitioners in selecting an appropriate SSL strategy based on various parameters. Finally, combining both *a contrario* and SSL paradigms has led to impressive performance for IRSTD.

Thèse préparée à l'ONERA - The French Aerospace Lab (Palaiseau) et au laboratoire SATIE (Université Paris-Saclay), avec un co-financement ONERA/Safran E&D.

Remerciements

Cette belle et riche aventure qu'est la thèse n'aurait sans doute jamais vu le jour si mon chemin n'avait pas croisé celui de mes deux mentors, Florin et Andrei. Je n'envisageais pas, au départ, de faire une thèse après l'école d'ingénieurs, mais cela s'est avéré être une des meilleures décisions de ma vie professionnelle. J'ai énormément appris et m'y suis sentie très épanouie, et je suis vraiment très heureuse de pouvoir rejoindre à mon tour la communauté de chercheurs. Alors rien que pour ça, un grand merci à vous deux.

Je tiens ensuite à exprimer ma gratitude aux membres du jury pour le temps consacré à l'évaluation de mes travaux, ainsi que pour leurs questions et remarques très pertinentes. Vos retours m'ont ouvert de nouvelles perspectives prometteuses que j'ai hâte d'explorer dans la suite de ma carrière.

Cette thèse n'aurait pas été possible sans le soutien de mes encadrants de l'ONERA, du SATIE et de Safran E&D. Merci Arnaud et Clara pour vos conseils et pour m'avoir permis de découvrir le milieu industriel. Merci Sidonie pour ton optimisme, ton soutien, ta confiance et ton aide précieuse, notamment dans le partage des références bibliographiques. Travailler avec toi a été un véritable plaisir, et j'ai beaucoup apprécié les discussions scientifiques passionnantes qu'on a pu avoir autour du sujet de thèse. Enfin, un immense merci à Sylvie, pour ton accompagnement à la fois bienveillant et exigeant. Ta rigueur scientifique et ta capacité à m'encourager à toujours aller plus loin ont été essentielles à la réussite de cette thèse et m'ont fait prendre en maturité scientifique. Tu es pour moi une grande source d'inspiration et j'espère suivre tes pas dans ma future carrière.

Je remercie également les équipes qui m'ont accueillie chaleureusement : celles de Safran, du SATIE, et surtout de l'ONERA. Merci aux équipes du DOTA et particulièrement à MPSO pour les échanges stimulants et les conditions de travail idéales. Un grand merci également à l'AMIAD pour son accueil chaleureux en fin de thèse et pour m'offrir l'opportunité de poursuivre ces travaux dans le cadre d'un post-doc.

J'ai bien évidemment une pensée particulière pour mes compères qui ont rendu ces trois années merveilleuses. Merci aux co-doctorants du SATIE pour les moments partagés, allant des soirées mémorables aux collaborations sportives et musicales. Et surtout merci à la grande équipe de thésards de l'ONERA, même si je ne suis pas une physicienne (n'en déplaise à Jo), je vous suis très reconnaissante pour tous ces beaux souvenirs que l'on s'est forgés ensemble, que ce soit lors des sessions karting, MK, Sham, ou des soirées raclette, karaoké (ou autres thèmes plus ou moins assumables). Merci aux "zoomers" d'avoir élargi mon vocabulaire d'une manière conséquente, merci également au meilleur bureau du 3ème étage (et

de tout le bâtiment) pour la bonne humeur, l'utilisation artistique des différentes IA, la boîte à Meuh et pour toutes ces chansons emblématiques qui ont rythmé nos journées. Merci au stagios, même si je n'ai jamais eu droit à mon thé quotidien, j'ai apprécié ta main d'œuvre efficace (et de grande qualité, il faut le dire !). Je ne citerai pas plus de noms puisque la liste serait trop longue, mais vous savez que vous avez tous une place spéciale dans mon cœur qui va au-delà de simples "collègues" (n'en déplaise cette fois-ci à Dumé). Merci à chacun d'avoir rendu cette période de ma vie si riche humainement, et je n'ai aucun doute sur le fait que la fin de la thèse ne marque en rien la fin de tout cela.

Enfin, je remercie mes proches, mes amis et ma famille, pour leur soutien indéfectible et leur amour tout au long de ces années, à tous moments. Vous avez contribué à rendre ma vie plus douce, et je vous en suis profondément reconnaissante.

Synthèse de la thèse

La détection de petits objets dans les images infrarouges (IR) est essentielle pour diverses applications de défense, notamment dans le renseignement militaire, la surveillance maritime et l'interception de missiles. Cette détection constitue la première étape des approches DRI (Détection-Reconnaissance-Identification), et il est crucial de minimiser les erreurs et les détections manquées. Pour de nombreuses applications militaires, il est également vital de détecter la cible dès que possible. Cependant, si la cible est située à plusieurs kilomètres de l'observateur, elle peut n'occuper qu'une infime portion de l'image. De plus, la réfraction atmosphérique et d'autres bruits peuvent masquer la cible dans des fonds complexes et texturés. Les cibles n'ont donc pas de structure spécifique (basse résolution) et présentent un faible contraste par rapport au fond. Ces conditions font de la détection de petites cibles dans des images infrarouges un défi majeur en vision par ordinateur.

Au cours des dernières décennies, de nombreuses méthodes ont été proposées pou répondre à cette problématique. L'avènement des méthodes d'apprentissage profond a récemment conduit à des avancées impressionnantes en matière de détection d'objets, grâce à leur capacité à extraire des caractéristiques non linéaires à partir de grandes quantités de données annotées. L'émergence récente de jeux de données de petites cibles en IR a permis le développement de détecteurs de petites cibles infrarouges basés sur l'apprentissage profond, qui ont surpassé les méthodes traditionnelles. Ce sont en particuliers les réseaux de segmentation sémantique qui sont à l'état de l'art pour la détection de petites cibles en IR.

Dans cette thèse, notre objectif est de proposer des méthodes innovantes et efficaces pour la détection de petites cibles qui se généralisent bien à des applications légèrement différentes, toutes dans le domaine de la détection de petits objets. Nous souhaitons développer des détecteurs capables de bien se généraliser à tout type de données (par exemple, RGB, IR, mono ou multispectral, etc.) sans considérer explicitement les propriétés physiques du domaine. De plus, notre approche doit fonctionner efficacement dans un environnement frugal, ce qui constitue une contrainte importante de l'application. Étant donné que les petites cibles sont de très basse résolution et que la plupart des jeux de données publics sont monospectraux, notre objectif est uniquement de détecter les cibles sans caractériser les objets.

Dans la littérature, nous avons identifié plusieurs faiblesses. Tout d'abord, les fonds fortement texturés peuvent entraîner de nombreux faux positifs. Ensuite, les étapes de sous-échantillonnage successives dans les architectures d'apprentissage profond conventionnelles entraînent une perte d'informations sur les très petits objets, ce qui se traduit par de nombreuses détections manquées. Un autre prob-

lème majeur est l'asymétrie de l'apprentissage induit par le choix de la fonction de coût du réseau, où les pixels cibles et les pixels de fond sont pénalisés (presque) de manière égale. Cela signifie qu'en raison du nombre limité d'exemples de cibles, le processus d'apprentissage est principalement guidé par les erreurs commises sur les pixels de fond. Des techniques telles que la pondération de la fonction de coût selon la classe du pixel ou la super-résolution ont été proposées pour résoudre le problème. Cependant, ces méthodes ne tirent pas parti du caractère « inattendu » des petits objets par rapport au fond, comme on pourrait le faire dans une approche de détection d'anomalies.

Dans la première partie de la thèse, nous proposons un nouveau paradigme d'apprentissage profond pour la détection de petits objets en tenant compte de leur caractère inattendu. Il repose sur un raisonnement *a contrario* et permet de dériver automatiquement un critère de décision en modélisant le fond à l'aide d'un modèle naïf et en détectant des structures ou des objets trop structurés pour apparaître « par chance » sous le modèle naïf. Le paradigme *a contrario* s'inspire des théories de la perception, en particulier de la théorie de la Gestalt. Cette dernière est basée sur le principe de Helmholtz, qui stipule qu'une grande déviation par rapport à un modèle aléatoire est probablement due à la présence d'une structure. Notre motivation pour utiliser de telles méthodes est qu'elles modélisent le fond, pour lequel nous disposons de nombreux échantillons, plutôt que les objets à détecter. Cela contourne ainsi le problème du déséquilibre des classes en se concentrant sur la classe « fond » et en effectuant la détection par rejet de son hypothèse naïve. Cette modélisation semble d'autant plus appropriée que les petits objets contiennent souvent très peu de structures géométriques, contrairement aux objets plus grands pour lesquels la littérature est très abondante. De plus, la formulation *a contrario* vise à minimiser le Nombre de Fausses Alarmes (NFA), permettant ainsi un meilleur contrôle de la précision. Nous proposons donc de guider l'apprentissage des réseaux de neurones en intégrant le critère *a contrario* dans la boucle d'apprentissage par le biais d'un module NFA spécifique. Ce dernier guide le réseau à extraire des caractéristiques de manière à ce que les caractéristiques de l'objet soient susceptibles de contredire l'hypothèse naïve du fond. Cela induit des propriétés intéressantes : 1) les résultats sont plus interprétables ; 2) le choix du seuil permet un contrôle plus intuitif du NFA. Le module NFA que nous développons dans cette thèse peut être intégré dans n'importe quel réseau de segmentation. Cependant, s'appuyer sur des réseaux de segmentation pour la détection d'objets peut entraîner une fragmentation des objets lors du réglage du seuil de binarisation de la carte de segmentation. Cela peut provoquer de nombreuses fausses alarmes et fausser les métriques de comptage d'objets. Les algorithmes de détection d'objets tels que Faster-RCNN ou YOLO réduisent ce risque en localisant explicitement les objets par la régression de boîtes englobantes,

bien qu'ils aient souvent du mal avec les petits objets. Peu d'études ont adapté ces détecteurs pour la détection de petites cibles, et aucune comparaison rigoureuse avec les méthodes de segmentation à l'état de l'art pour la détection de petites cibles n'a été réalisée. Nous proposons donc d'intégrer les critères de décision *a contrario* dans la tête de détection YOLO. Plus précisément, nous explorons deux formulations *a contrario* différentes et démontrons leurs avantages par rapport à une version classique de YOLO. L'intégration de notre critère *a contrario* dans un YOLO a permis de réduire significativement l'écart de performance observé entre les réseaux de type YOLO et ceux de segmentation à l'état de l'art pour la détection de petites cibles.

Dans la deuxième partie de ce manuscrit, nous nous concentrons davantage sur les conditions d'apprentissage difficiles dues à la rareté des données annotées. En effet, ces conditions conduisent généralement à une représentation médiocre des données dans l'espace latent, car les réseaux de neurones nécessitent des milliers d'échantillons pour apprendre à extraire des caractéristiques pertinentes. Notre objectif est donc de compenser la frugalité des données en tirant parti de meilleures représentations ou de connaissances préalables. La pratique courante consiste à initialiser les modèles avec des poids pré-entraînés sur des jeux de données classiques et de grande taille, comme ImageNet. Bien que l'utilisation de poids pré-entraînés sur le dataset ImageNet de manière supervisée soit une approche standard, cela peut être une stratégie sous-optimale, en particulier lorsqu'on considère un grand écart entre les domaines spectraux (par exemple, visible et infrarouge thermique). Par conséquent, les méthodes de pré-entraînement non supervisé, notamment basées sur de l'apprentissage auto-supervisé (SSL), deviennent de plus en plus populaires. Le SSL constitue un domaine de recherche particulièrement actif. Il repose sur une tâche prétexte capable de générer sa propre vérité terrain, ce qui permet au réseau d'apprendre des invariances ou des caractéristiques pertinentes pour la tâche finale. Plusieurs stratégies pour sélectionner des tâches prétextes ont été proposées dans la littérature, initialement conçues pour des tâches de classification et adaptées au fur et à mesures aux tâches de prédiction dense ou locale (par exemple, segmentation, détection d'objets) et aux architectures de réseaux récentes comme les Vision Transformers. Ces méthodes ont montré des performances impressionnantes, par exemple dans la segmentation d'objets non supervisée. Cependant, leur efficacité a principalement été démontrée pour la détection de grands objets. Cela soulève la question suivante : ces méthodes à l'état de l'art sont-elles réellement adaptées à la détection de petits objets ? Dans ce manuscrit, nous commençons par passer en revue les principales méthodes historiquement orientées vers la classification et présentons leurs caractéristiques. Ensuite, nous nous concentrons sur la détection d'objets, en présentant des méthodes spécifiquement adaptées à ces tâches et en réalisant plusieurs benchmarks.

Nous commençons par évaluer la performance de différentes stratégies SSL sur le dataset COCO, en portant une attention particulière aux métriques obtenues sur les petits objets (bien qu'ils soient encore sensiblement plus grands que les petites cibles). Ensuite, nous considérons la détection de véhicules à partir d'images aériennes, ce qui implique de gérer des objets de plus petite taille. Ce jeu de données inclut à la fois des images RGB et IR, permettant d'amener le sujet du transfert de connaissance inter-domaine, comme par exemple du domaine RG vers l'IR. Nous examinons ensuite quelles méthodes sont les mieux adaptées pour l'entraînement sur un grand jeu de données intra-domaine (données IR dans notre cas) qui n'a peut-être pas été nettoyé (par exemple, redondance d'images, hétérogénéité sémantique). Enfin, nous évaluons ces différentes méthodes de pré-entraînement sur notre tâche d'intérêt, à savoir la détection de petites cibles en IR. Dans cette deuxième partie de la thèse, notre approche est observationnelle : nous analysons le comportement des méthodes existantes, notons les différences dans les résultats et tentons d'identifier les composantes responsables de ces différences. Cette approche d'analyse structurée fournit des perspectives pour de futures recherches. Enfin, la combinaison des paradigmes *a contrario* et auto-supervisé conduit à des résultats impressionnants pour la détection de petites cibles en IR.

Contents

List of Figures

List of Tables

Acronyms

AI	Artificial Intelligence
AIS	Automatic Identification Systems
AP	Average Precision
CNN	Convolutional Neural Networks
DBT	Detect Before Track
DL	Deep Learning
EMA	Exponential Moving Average
FA	False Alarm
FCN	Fully Connected Networks
FIR	Far InfraRed
FN	False Negative
FP	False Positive
FPN	Feature Pyramid Networks
IM	Image Modelling
IoU	Intersection over Union
IR	InfraRed
IRSTD	InfraRed Small Target Detection
LWIR	Long-Wavelength InfraRed
MAE	Masked Auto-Encoder

mAP mean Average Precision

MC Multi-Channel

MIM Masked Image Modelling

MWIR Mid-Wavelength InfraRed

NFA Number of False Alarms

NIR Near InfraRed

NMS Non-Maximum Suppression

NN Neural Networks

ResNet Residual Network

ROI Region Of Interest

RPN Region Proposal Network

SOTA State-Of-The-Art

SPIE Society of Photo-Optical Instrumentation

SSL Self-Supervised Learning

SVM Support Vector Machine

SWIR Short-Wavelength InfraRed

TBD Track Before Detect

TP True Positive

ViT Vision Transformer

YOLO You Only Look Once

Chapter 1

Introduction

1.1 Challenges behind Infrared Small Target Detection

Accurate detection of small objects in InfraRed (IR) images is essential in various defense applications, including military intelligence, maritime surveillance, and missile interception. It also finds applications in civilian fields such as medical diagnosis, early fire detection, and video surveillance. This detection is the first step in Detection-Recognition-Identification (DRI) approaches, making it a critical stage where errors and missed detections must be minimised.

Many civilian and military applications use the principle of infrared radiation to identify objects in an optical scene. Indeed, all objects reflect and emit infrared radiations that are characteristic of their physico-chemical properties and temperature. When an object has a temperature that is significantly different from its surroundings (e.g., an aircraft in flight), it can thus be observed in an infrared image. Additionally, IR imaging has benefits over active radar systems, such as good portability and low sensitivity to strong electromagnetic interference and stealth, as it is passive.

The infrared domain includes wavelengths from 750 nm to 1 mm, divided into five categories:

- **Near InfraRed (NIR)**: 750 nm to 1.4µm, used in low light conditions (e.g., night vision goggles). It provides an image that is close to a visible one.
- **Short-Wavelength InfraRed (SWIR)**: 1.4 to 3µm, captures light reflected or absorbed by objects. Known as reflected IR, this domain is less affected by fog or pollution, offering a significant advantage over visible cameras.

- **Mid-Wavelength InfraRed (MWIR)**: 3 to 8µm, detects thermal emission of objects but struggles with fog or pollution. It consists of both reflected and thermal infrared, and it is ideal for detecting aircrafts.

- **Long-Wavelength InfraRed (LWIR)**: 8 to 15µm, preferred for outdoor surveillance and military missions as it is less sensitive to the thermal noise caused by the surrounding environment. It is known as thermal infrared.

- **Far InfraRed (FIR)**: 15µm to 1 mm, used in astronomy applications (black-body radiations).

Infrared target detectors are primarily applied to SWIR, MWIR, or LWIR images, and most public datasets are composed of monospectral images. However, for target recognition and identification, infrared multispectral or hyperspectral images should be considered, as the variety of IR bands can enable discrimination between detected targets.

For many military applications and systems (e.g., anti-collision systems), detecting the target as soon as possible is essential. However, if the target is kilometers away from the observer, it may occupy a very small portion of the image. Furthermore, atmospheric refraction and other noises tend to drown the target in complex backgrounds, thus greatly reducing its signal-to-noise ratio (SNR). Targets thus have no specific structure (low-resolution) and present low contrast with respect to the background. These conditions make InfraRed Small Target Detection (IRSTD) a significant challenge in computer vision. We can illustrate these difficulties with Figure 1.1. It shows three real infrared scenes containing targets, as well as their corresponding 3D surface representation. The position of the targets is indicated by red rectangles. Figure 1.1a) represents a simple case, where the target is not drown into noise and has a high intensity value compared to the neighbourhood. Figure 1.1b) illustrates an intermediate case. Indeed, the target has a lower intensity and a lower contrast compared to the surrounding background. The intensity peak is already more difficult to detect. Finally, Figure 1.1c) represents a complicated case: the target is particularly small and has a low intensity value. Furthermore, as the background is complex and highly textured, the target is drown into clutter noise, represented by relatively high intensity peaks. This makes the background subtraction difficult, and can also lead to many false alarms. The design of an infrared small target detector that is robust towards false positives and has a high generalisation ability is thus all the more important. Another important constraint of our application is the scarcity of both samples *and* annotated data. Due to the rarity of targets, we only have access to a limited number of pixel samples belonging to the target class. Furthermore, since we often work with specialised sensors (e.g., thermal infrared, hyperspectral

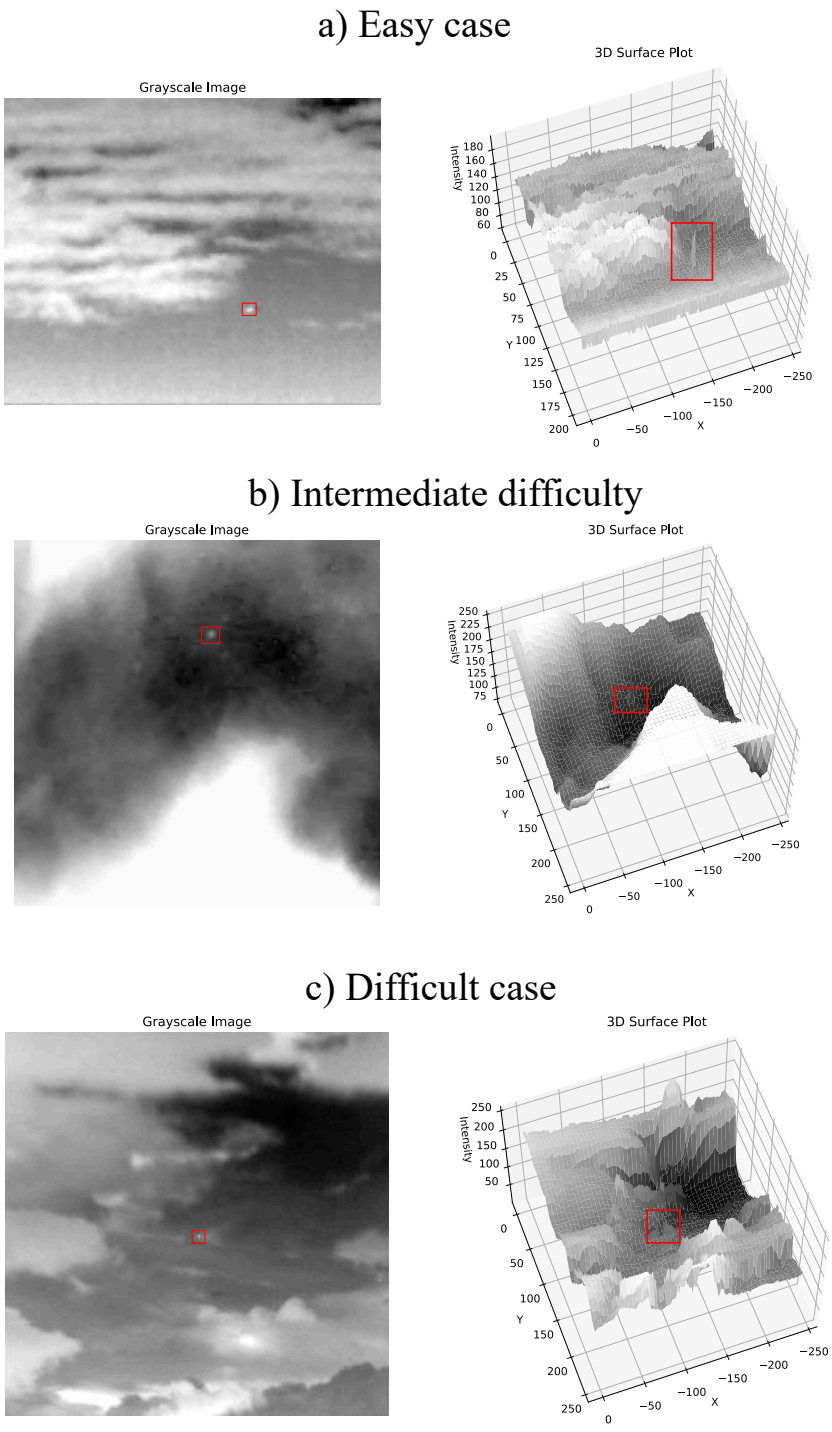


Figure 1.1: 3D representation of some infrared images. The targets are framed in red.

data) and given the high cost of annotation, the number of annotated samples is also limited.

1.2 IRSTD methods still need to be improved

Over the past decades, many IRSTD methods have been proposed. Among them, Detect Before Track (DBT) and Track Before Detect (TBD) methods have been designed for IRSTD on video sequences. On the one hand, DBT methods first rely on single-frame detection [16], where candidate targets are selected in each frame, and then temporal information confirms the final targets based on trajectory estimation. On the other hand, TBD methods directly exploit temporal information for detecting the targets [17]. However, they are more computationally expensive than DBT methods, and require large storage capacity. Moreover, there are cases where short-term temporal information is inefficient for target detection. For example, when a target approaches the sensor head-on, it initially appears as a very small dot, making it challenging to quantify its size evolution at the beginning of the sequence. This scenario highlights the importance of single-frame IRSTD. This, coupled with the rise of very efficient deep learning-based detectors, encourages research into DBT detection, and in particular single-frame infrared detection of small targets.

Before the advent of deep learning and the publication of large-scale IRSTD datasets, many model-driven methods were proposed. These include, for example, spatial [18, 19] and frequency domain [20] filtering for background removal, as well as local contrast estimation [21, 22, 23] (which assumes that the target and its surrounding environment present abrupt changes in terms of pixel intensity). Since model-driven methods are beyond the scope of this thesis, we refer the reader to surveys [24, 25] for further details. An obvious drawback of these methods is that they rely on heavy parameter tuning, and thus present a very weak generalisation ability to other sensor images or to different backgrounds.

The rise of Deep Learning (DL) methods has recently led to impressive advances in object detection, thanks to their ability to learn to extract non-linear features from large amounts of annotated data. The recent emergence of IRSTD datasets, such as the pioneering work by Wang et al. [9] in 2019, has led to the development of DL-based infrared small target detectors, which have outperformed traditional methods. In contrast to model-driven approaches, these methods show great generalisation ability, require little parameter tuning, and achieve better performance. They also have short inference times, making real-time detection possible. Semantic segmentation neural networks (NN) are the most widely used networks for IRSTD and lead to State-Of-The-Art (SOTA) performance on several IRSTD benchmarks. These include ACM [6], LSPM [26] and one of the recent

SOTA methods, namely DNANet [8], which consists of several nested UNets and a multiscale fusion module that helps the segmentation of small objects with various sizes. Most of DL-based methods are designed to be independent of the physical properties of materials, allowing them to work with different types of data such as IR, RGB, and multi/hyperspectral data.

In this doctoral thesis, our objective is to propose innovative and efficient methods for small target detection that generalise well to slightly different applications, all within the domain of small object detection. As with the other DL-based methods for IRSTD, we aim to develop detectors that can generalise well to any type of data (e.g., RGB, IR, mono or multispectral, etc.) without explicitly considering the physical properties of the domain. Additionally, our approach must operate effectively in a frugal setting, which is a significant constraint of the application. Since the small targets are of very low-resolution and that most public datasets are monospectral, our focus is solely on detecting the targets without characterising the objects.

In the literature, we have identified several weaknesses. First, highly textured backgrounds can lead to many false positives. Second, the successive downsampling steps in conventional deep learning architectures lead to a loss of information about very small objects, resulting in many missed detections. Another major issue is the symmetric learning approach, where target pixels and background pixels are penalised (almost) equally. This means that, because of the limited number of target examples, the training process is primarily driven by the errors made on the background pixels. Techniques such as loss weighting (e.g., focal loss [27]) or super-resolution [28] have been proposed to address the issue. However, these methods do not take advantage of the *unexpectedness* of small objects with respect to the background, as one could do in an anomaly detection approach with, for example, one-class classifiers [29] that discriminate small objects as *unexpected* patterns with respect to the background. Such a criterion can efficiently reduce the number of false alarms induced by the background, thus allowing for a better balance between precision and detection rate.

1.3 Organisation of the manuscript

In the first part of the thesis, we propose a new deep learning paradigm for detecting tiny objects by considering their unexpectedness. It relies on *a contrario* reasoning, introduced by [30]. These methods allow us to automatically derive a decision criterion by modelling the background using a naive model and detecting structures or objects as too structured to appear 'by chance' under the naive model. They draw inspiration from theories of perception, in particular the Gestalt theory [12]. The latter is based on the Helmholtz principle, which states that a

large deviation from a random pattern is likely due to the presence of a structure. Our motivation for using such methods is that they model the background, for which we have a lot of samples, rather than the objects to be detected. It thus circumvents the problem of class imbalance by focusing on the background class and performing detection by rejection of its distribution hypothesis. Such a modelling appears even more appropriate as tiny objects often contain very few geometric features, unlike larger objects for which the literature is very extensive. Moreover, the *a contrario* formulation aims at minimising the Number of False Alarms (NFA), defined in Chapter 3, thus allowing for a better control of the precision.

In the literature, the *a contrario* decision is applied either directly on natural images or after extracting features from the image by traditional image processing methods. This filtering step can be replaced by Neural Networks (NN). Indeed, when looking at the feature maps of a NN trained for detecting objects, the objects to be detected stand out against a background made of noise. [31] applied, as a post-processing step, *a contrario* detection on some feature maps obtained by a NN. We follow the same approach and apply an *a contrario* test on the feature maps extracted by conventional segmentation networks trained on IRSTD datasets. This first stage, presented in the second part of Chapter 3, is intended as a proof of concept of the use of the *a contrario* criterion in the detection of small targets.

However, it is clear that applying the *a contrario* testing on the feature maps extracted by a neural network while being agnostic to the future detection criterion appears suboptimal. Indeed, the feature map statistical distribution may not match the naive assumption made on the background when applying *a contrario* decision. We therefore propose to guide the NN training by including the *a contrario* criterion in the training loop through a specific NFA module. The latter guides the network to extract features in such a way that the object features will be likely to contradict the naive hypothesis made on the background. This induces interesting properties: 1) the results are more interpretable; 2) the threshold choice allows for a more intuitive control of the NFA. The NFA module that we develop in this thesis can be integrated into any segmentation NN, and can even take advantage of multi-scale information if the backbone allows for it. The methodology as well as the results obtained for different use cases are presented in Chapter 4.

However, relying on segmentation NN for object detection can lead to object fragmentation when tuning the threshold for binarising the segmentation map. This can cause many false alarms and distort object counting metrics. Object detection algorithms such as Faster-RCNN [3] or You Only Look Once (YOLO) [4] reduce this risk by localizing objects through bounding box regression, although they often struggle with small objects. Few studies have adapted such detectors

for IRSTD [32], and no rigorous comparison with SOTA IRSTD methods has been made. We therefore propose to integrate *a contrario* decision criteria into the YOLO detection head. Specifically, we explore two different *a contrario* formulations (one at pixel-level, and the other at the object-level) and demonstrate their advantages over a robust YOLO baseline in Chapter 5. This confirms the relevance of the *a contrario* paradigm for the detection of small targets, even if the integration of the pixel-level *a contrario* criterion introduced in Chapters 3 and 4 remains the new SOTA approach.

In the second part of this manuscript, we address challenging training conditions caused by the scarcity of samples and annotated data. Indeed, such conditions generally lead to poor data representation in the latent space, as neural networks require thousands of samples to learn to extract meaningful features. Our objective is thus to compensate for data frugality by leveraging better representations or prior knowledge. Common practice involves initialising models with weights pre-trained on classic and large computer vision datasets like ImageNet [33]. While using weights pre-trained on ImageNet dataset in a supervised way is a standard approach, this may be a sub-optimal strategy especially when considering a large gap between the spectral domains (e.g., visible and thermal-IR). Therefore, unsupervised pre-training is becoming more and more popular. An advantage of such pre-training is that it does not rely on annotated data, allowing pre-training on large amounts of data across different domains. In this way, the massive amount of unlabelled data provided by the large number of sensors can be exploited. It also opens up new possibilities for applications where the amount of annotated data is too small, and where deep learning methods are not yet viable.

Self-Supervised Learning (SSL) is a SOTA approach for performing unsupervised pre-training on large unlabelled datasets, and is a particularly active area of research. It relies on a pretext training task able to generate its own ground truth (e.g., pseudo-labels), and such a strategy helps the network to learn invariances and latent patterns in the data. Several strategies for selecting pretext tasks have been proposed in the literature, initially designed for classification tasks and now increasingly adapted to dense or local prediction tasks (e.g., segmentation, object detection) and recent network architectures like Vision Transformer (ViT) [1]. These methods have shown impressive performance, for example in unsupervised object segmentation (e.g., Leopart [34]). However, their effectiveness has primarily been demonstrated on large objects. This raises the following question: are these SOTA methods actually suitable for small object detection?

In this manuscript, firstly, we review the main methods historically oriented towards classification and their characteristics in Chapter 6. Then, we shift our focus to object detection in Chapter 7, presenting methods specifically adapted to these tasks and conducting several benchmarks. We start by evaluating the

performance of different SSL strategies on the classic COCO dataset [35], paying a particular attention to small objects (although they are still significantly larger than small targets). Next, we consider vehicle detection from aerial images, which involves handling smaller objects and different camera perspectives. This dataset includes both RGB and IR images, facilitating a smooth transition to out-domain transfer learning. We then investigate which methods are best suited for training on large in-domain dataset (IR data in our case) that may not have been cleaned (e.g., image redundancy, semantic heterogeneity). Finally, we evaluate these different pre-training methods on our task of interest, namely infrared small target detection. In this second part of the thesis, our approach is observational: we analyse the behaviour of existing methods, note the differences in results, and attempt to identify the components responsible for these differences. This structured analysis approach provides valuable insights and perspectives for future research. Last but not least, combining both *a contrario* and self-supervised paradigms lead to new state-of-the-art results for infrared small target detection.

Chapter 2

Related works

In this chapter, we provide foundational material essential for understanding the concepts discussed later in the manuscript. We begin with an introduction to object segmentation and detection methods, followed by a presentation of key state-of-the-art approaches for infrared small target detection. Lastly, we introduce the datasets that will be used to train and evaluate our methods.

2.1 Object segmentation and detection methods

Object detection and segmentation are essential tasks in the field of computer vision, with a wide range of applications, including medical imaging, scene analysis (e.g., for autonomous driving), or video surveillance. These tasks consist of identifying objects within an image or video and delineating their boundaries, which is a first step in understanding visual scenes. In recent years, the advent of Convolutional Neural Networks (CNN) and other deep learning paradigms has enabled the development of complex models capable of real-time object detection with impressive performance. In the following, we will introduce how feature extraction is performed by CNN and more recent deep learning paradigms such as ViT. Then, we will present object detectors and common segmentation networks. Finally, we will introduce the main metrics used for evaluating object detectors.

2.1.1 Deep learning-based feature extraction

Object detection and segmentation networks consist of two key elements: the encoder, which allows for feature extraction (i.e., extracting relevant patterns in the image), and a detection or segmentation head. Before the advent of deep learning, feature extraction in computer vision was performed using a variety of hand-crafted techniques. These methods aimed to detect and describe important characteris-

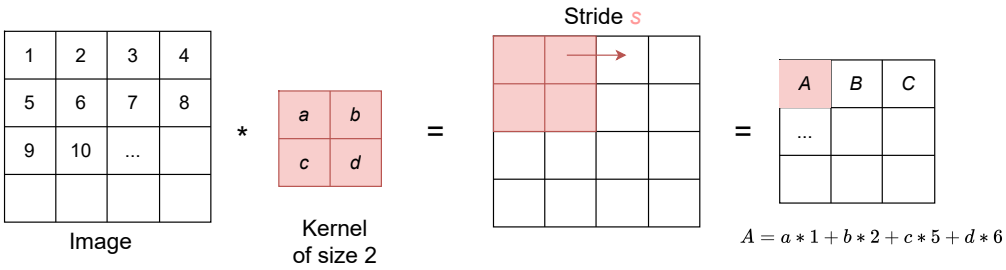


Figure 2.1: Illustration of a convolution of a 4×4 image with a 2×2 kernel, with a stride of 1 and no padding ($p = 0$).

tics in images, such as edges, textures or shapes, which could then be used for tasks such as classification, or object recognition. However, they required careful tuning and were often limited by their inability to generalise across different tasks and datasets. The advent of deep learning has revolutionised feature extraction by allowing models to learn features directly from data, resulting in more robust and general representations. In this section, we will introduce the basics of deep learning-based feature extraction. We start with a classic and fundamental network in deep learning, namely CNN, and then go on with a recent and promising architecture, namely ViT.

Convolutional neural networks – Convolutional Neural Networks are a class of deep neural networks commonly used for image feature extraction, and are composed of several convolution layers. The latter are based on convolution operations, which consist in performing a dot product between two matrices: one matrix is a set of learnable parameters called a kernel or a filter, and the other one is a small patch (called the neuron's receptive field) from the original image or feature map, with the same spatial dimension as the kernel. Common sizes $k \times k$ for the kernel are 3×3, 5×5, or 7×7. As illustrated on Figure 2.1, the kernel is slid over the input image/feature map of spatial size $H \times W$. The sliding gap is called the stride s, and it is common to use a stride equal to 1 or 2 (e.g., to downsample the input feature map by a factor of 2). Note that the input image can be padded (usually with 0 at the borders, with a margin of p) in order to control the spatial dimensions of the output feature map. The convolution outputs a feature map of spatial size $\lfloor \frac{H+2*p-k-2}{s} + 1 \rfloor \times \lfloor \frac{W+2*p-k-2}{s} + 1 \rfloor$ and with the same depth as the kernel depth. A bias term is also usually added. In the case of an input with multiple channels (e.g., RGB channels), the convolution is performed separately for each channel, and the results are summed over the channels. After the convolution operation, an activation function (e.g., ReLU) is applied in order to introduce nonlinearity and to help the network learn complex patterns.

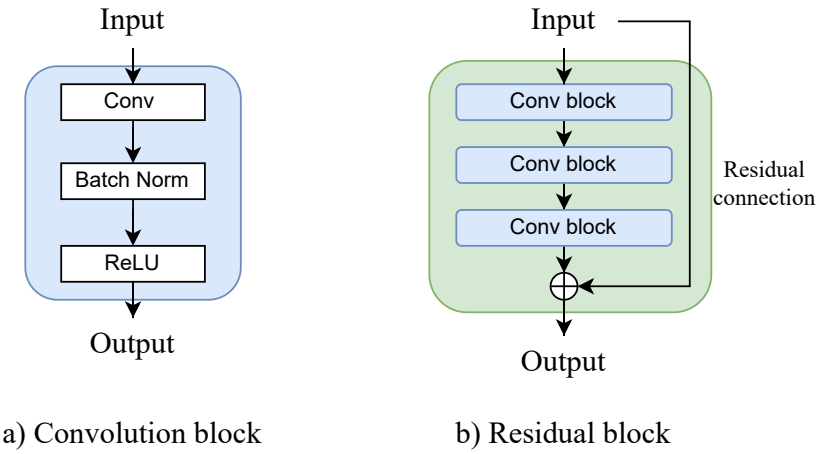


Figure 2.2: Illustration of a) a convolution block, and b) a ResNet bottleneck with residual connection.

Abstract features can thus be extracted from images by stacking multiple convolutional layers, resulting in a CNN. One of the first introduced CNNs is AlexNet [36], which has demonstrated high performance over traditional methods on the ImageNet dataset [33], a large-scale classification dataset. The hierarchical structure is obtained by using several downsampling steps, which can be performed either by using a stride of size 2 in the convolution, or by using pooling layers. The latter reduce the dimension of the feature map by combining the values included within a small window that is slid over the whole feature map with a stride s. Typically, a window of size 2×2 is considered to reduce the feature map by a factor of 4. Max pooling operation is often used: it returns the maximum value within a given window.

Building on the success of AlexNet, researchers have developed a series of increasingly sophisticated CNNs for feature extraction. Among them are Residual Network (ResNet) [37]. The latter address some optimisation difficulties (e.g., vanishing gradients) observed when training very deep networks by introducing residual connections. More specifically, a ResNet is mainly composed of residual blocks, which consist of a set of convolutional layers where the input is directly added to the output of the layer stack, forming a shortcut connection, as illustrated on Figure 2.2b). Then, the result of the addition is passed through an activation function (e.g., ReLU).

Among all the ResNet variants (which differ in the number of convolutions), the ResNet-50 is the one that is most commonly used. It consists of 50 convolutional layers distributed over 16 residual blocks. Table 2.1 gives details of its architecture.

After the last convolutional block of AlexNet or any ResNet, it is possible to

layer name	ResNet-50
conv1	7×7, 64, stride 2 3×3 max pool, stride 2
conv2_x	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$
conv4_x	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$
conv5_x	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$

Table 2.1: Architecture of a ResNet-50. Each bracket represents a building block, illustrated in Figure 2.2a). The downsampling is performed by conv3_1, conv4_1, and conv5_1 with a stride of 2.

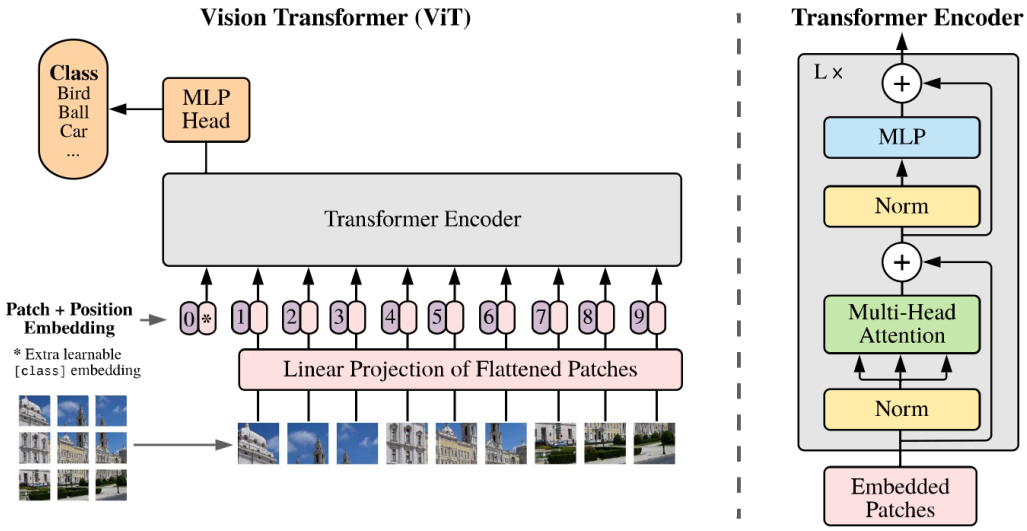


Figure 2.3: ViT architecture. Figure taken from [1] (Fig. 1).

add a fully-connected layer (MLP) along with a specific activation function (e.g., Softmax) to perform classification. These components form the classification head. However, in the context of object segmentation or detection, we drop the classification head and refer to the resulting network as the encoder or the backbone. A detection or segmentation head can then be added at the end of the encoder to perform object detection or segmentation. While AlexNet and ResNets are fundamental encoders in the field of CNNs, concurrent works such as DenseNet [38] or EfficientNet [39] have introduced significant innovations and have increased the performance for several tasks. Even more recently, ConvNeXts [40] have been introduced and outperform any previous CNN-based encoder. Their design is inspired by the components used in Vision Transformers, which we will present in the next section.

Vision Transformers – Recently, attention mechanisms have emerged as powerful mechanisms for analysing a scene. Inspired by human perception, they allow for a dynamic weighting of features according to their relevance to a given task. One innovative architecture that has revolutionised image understanding is the ViT [1], inspired by the success of Transformers in natural language processing. By dividing images into smaller patches and using self-attention mechanisms, a ViT explicitly captures long-range dependencies within images, achieving impressive performance in various computer vision tasks. Let us briefly introduce the main components of ViT encoder, whose architecture is presented in Figure 2.3:

- **Patch and position embeddings** – First, as illustrated in the left part of

Figure 2.3, the input image is divided into fixed-size patches (e.g., 16×16 pixels). Each patch is then flattened and transformed using a learnable linear projector (e.g., a convolution). The obtained embedding is called a "token". Position embeddings are added to the patch embeddings at the input stage to preserve the spatial position of a given patch.

- **Multi-head self-attention** – Then, the embeddings are passed through the transformer encoder, illustrated on the right side of Figure 2.3. A self-attention layer computes the attention scores between all pairs of patches in order to model their relationships. For a given query patch Q, the attention scores are computed based on the similarity between this patch and all the other patches in the image, called the key K. Then, the self-attention consists of a weighted sum of all patch embeddings V, where the weights are defined by the previously computed attention scores. The mathematical formulation of this operation is the following:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{2.1}$$

  where d_k is the dimension of the keys. We refer the reader to the fundamental papers [41, 1] for more details. Note that the self-attention block is preceded by a normalisation layer (e.g., Layer Norm), and has residual connections. Several self-attention layers are computed in parallel, leading to the so-called "multi-head" self-attention.

- **Feedforward neural network** – The output of the self-attention layer is then passed through a feedforward layer in order to introduce non-linearity and to learn contextual representations of image patches. It consists of a fully-connected layer followed by an activation function. As for the self-attention block, it is preceded by a normalisation layer and has residual connections.

There are several key differences between ViTs and CNNs. First, while CNNs process individual pixel values directly, ViTs divide the image into patches and transform them into tokens. Second, ViTs explicitly capture some global relationships through self-attention layers. In contrast, CNNs are better at extracting local features, although they can also somehow model large-scale relationships through multi-scale feature maps obtained from the downsampling steps. Finally, unlike the original ViTs, CNNs have a hierarchical architecture with multiple scales, which benefits the detection of objects at different scales.

A notable ViT variant that introduces hierarchical feature maps is the Swin-Transformer [2]. This hierarchical architecture is achieved through the patch merging process, as illustrated in Figure 2.4: the input image of size $H \times W$ is first

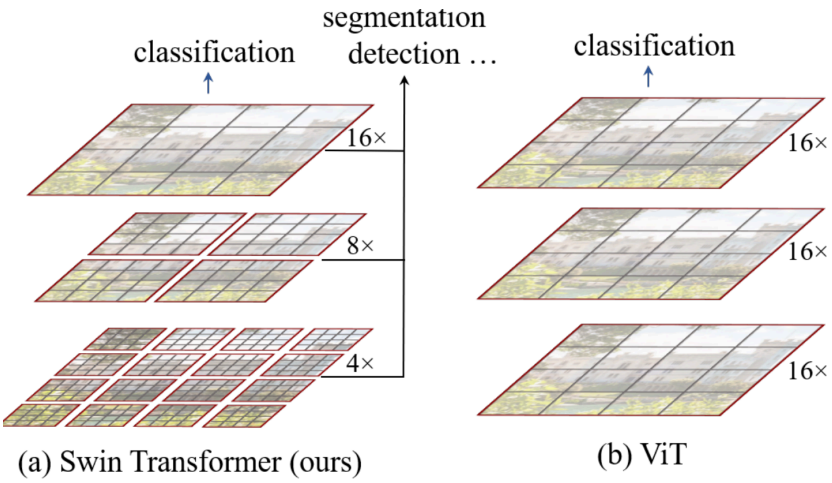
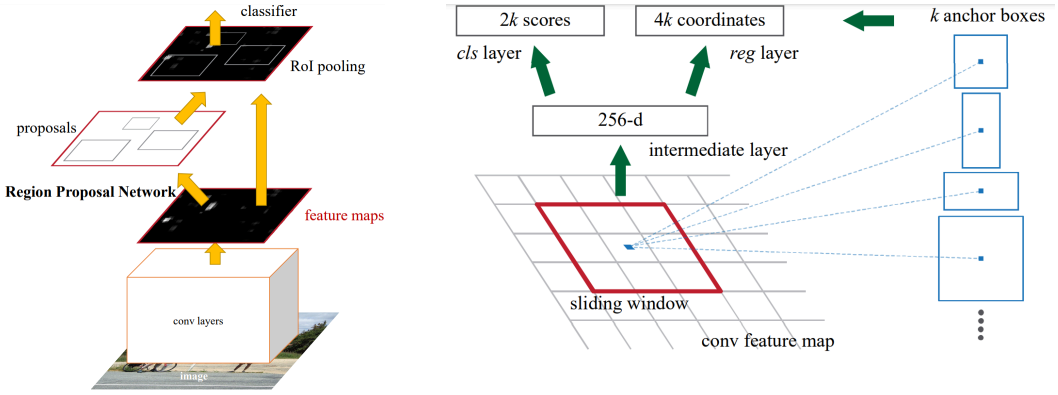


Figure 2.4: a) Patch merging process introduced in Swin Transformers, and comparison with b) a ViT architecture. Figure taken from [2] (Fig. 1). Copyright ©2021, IEEE.

divided into $\frac{H}{4} \times \frac{H}{4}$ non-overlapping windows. These windows are further divided into patches, and a classical ViT module is applied on each window. Then, the features from each non-overlapping group of 2×2 neighbouring windows are concatenated and linearly transformed, reducing the resolution by a factor of 2 in each spatial dimension (i.e., resulting in $\frac{H}{8} \times \frac{W}{8}$ patches). This process is repeated twice in order to obtain output resolutions $\frac{H}{16} \times \frac{W}{16}$ and $\frac{H}{32} \times \frac{W}{32}$. Note that the number of patches per window is constant for each resolution scale, allowing self-attention to be computed on smaller patches while maintaining computational complexity. Another key component of Swin-Transformers is the shifted window partitioning, which computes self-attention on two partitioning configurations (as detailed in Fig. 2 of [2]), introducing cross-window connections and improving performance on several dense prediction task benchmarks.

Another way to obtain ViT-based hierarchical architectures is to combine ViTs and CNNs, as done in the TransUnet [42] encoder. In the latter, a CNN produces feature maps at the scales $\frac{1}{2}$, $\frac{1}{4}$ and $\frac{1}{8}$. Then, ViT layers are added to produce a feature map at scale $\frac{1}{16}$. The authors of TransUnet [42] have found that such a design works better than using a pure non-hierarchical ViT encoder, especially for capturing smaller details.

Now that we have briefly presented the most emblematic CNN and ViT-based encoders, we will introduce some conventional object detection and segmentation architectures.



a) Faster R-CNN framework b) Region Proposal Network (RPN)

Figure 2.5: Illustration of a) the Faster R-CNN framework and b) the Region Proposal Network (RPN). Figures taken from [3] (Fig. 2 and Fig. 3).

2.1.2 Object detection networks

Object detection is the task of detecting objects of interest within an image and identifying their locations with bounding boxes, which consists of rectangular boxes defined by the spatial coordinates (x, y) of a specific point of the box (e.g., the top-left corner or the centre of the box) and the box size $(\text{width}, \text{height})$. It also provides a classification label for each detected object. Several types of deep learning approaches have been proposed for such a task, which we can group into two categories: two-stages and single-stage object detectors.

Two-stage architectures – One of the fundamental two-stage object detectors is R-CNN [43]. Introduced in 2014, it consists of 1) a region proposal step that relies on selective search [44], and 2) a feature extraction performed on each Region Of Interest (ROI) by a CNN such as AlexNet [36]. A linear Support Vector Machine (SVM) is then applied to provide classification scores, and the bounding box location is refined using a bounding-box regressor. Note that the extraction of features for each ROI (step 2) is computationally expensive and not optimal. Indeed, the feature extractor is applied several thousands of times for a single image. This is why Fast R-CNN [45] has been introduced: in this new version of R-CNN, the coordinates of the ROIs computed in the first step are projected on the feature maps extracted by the CNN (i.e., there is only one feature extraction per image) in order to obtain a feature vector for each ROI. However one problem remains: the selective search algorithm relies on traditional image processing methods and has slow execution times. Faster R-CNN [3] thus substitutes the selective search algorithm by a Region Proposal Network (RPN). As shown on Figure 2.5a, the latter generates region proposals directly from the convolutional

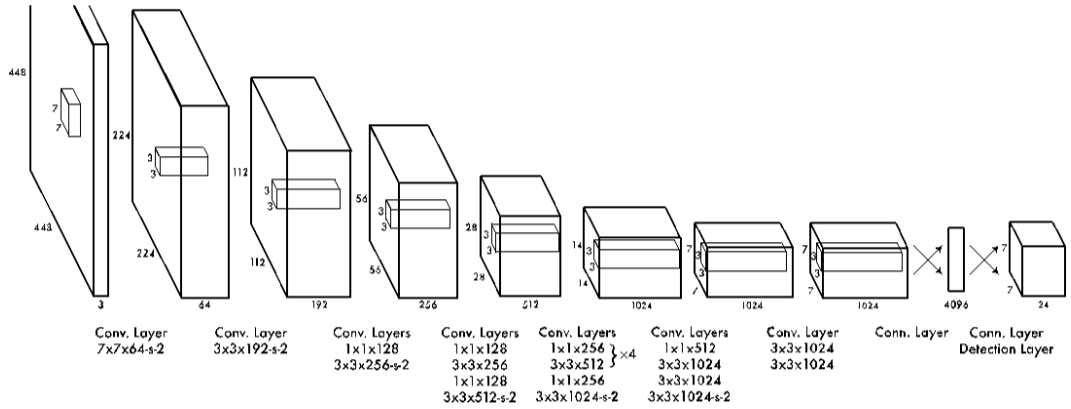


Figure 2.6: Illustration of the YOLO (first version) architecture. Figure taken from [4] (Fig. 3). Copyright ©2016, IEEE.

feature maps, which speeds up the detection process. RPN relies on the use of anchors (Figure 2.5b), which are pre-defined anchor boxes placed at different positions in the image and with different sizes and aspect ratios, allowing for the detection of objects of different shapes and sizes. For each anchor (which location is defined by a sliding window on the input image), the RPN makes k bounding-box proposals. Each bounding-box proposal consists of four refined coordinates (relative to a given anchor), and a classification score that indicates the presence of an object (also known as *objectness* score). Note that this multiple prediction process leads to redundant predictions. A post-processing technique called Non-Maximum Suppression (NMS) is thus applied. It consists of selecting the boxes having the highest *objectness* score, and suppressing the other boxes that have a high Intersection over Union (IoU) (introduced in Section 2.1.4) with it. Faster R-CNN has been further improved, for example by using other encoders such as ResNets and ViTs, or introducing multi-scale features through the use of Feature Pyramid Networks (FPN) [46].

Single stage architectures – Single stage architectures consist of using a single neural network to predict together bounding box coordinates, objectness and classification scores. YOLO [4] is the first single-stage object detector introduced in the literature, and is illustrated in Figure 2.6. Concretely, it divides the image into a grid and predicts the probability (denoted as the objectness score) for any given grid cell to contain an object. It also predicts the refined bounding box coordinates as well as the classification score of the object if it exists. Although its execution time was very fast, the first version of YOLO performed worse than the two-stage detectors, especially for small objects. Indeed, if the object to detect is too small, it may occupy only a small portion of a grid cell, making it difficult for

YOLO to detect it accurately. Concurrently, Single Shot MultiBox Detector [47] (SSD) was proposed. Unlike the first version of YOLO, SSD predicts the objects using multiple anchor boxes on multi-scale feature maps. Although it is slower, it performs better at detecting small objects thanks to the multi-scale feature maps.

Other versions of YOLO have been proposed to address these issues. For example, YOLOv3 [48] adds a FPN in the architecture and performs the detection at each scale in order to improve the detection of objects with various size. Furthermore, it relies on a more robust backbone, namely Darknet-53. Anchor boxes and batch normalisation have also been introduced since YOLOv2. Some of the latest versions of YOLO, such as YOLOv7 [49], YOLOv8 [50] or the recent YOLOv9 [51], lead to competitive detection performance on several famous computer vision benchmarks, while also improving the execution speed. Note that tiny versions of YOLO with fewer convolutional layers have also been proposed. So far, YOLO framework is one of the most widely used object detectors as it leads to great performance in various applications, with low execution time.

With the advent of Vision Transformers, new object detectors have been proposed, such as DETR [52]. The latter combines a CNN and a transformer-based encoder-decoder architecture. It also simplifies the detection pipeline by removing the need for NMS or spatial anchors through the use of a bipartite matching loss. Despite these advantages, DETR suffers from low training convergence and high computational cost. A recent version, namely RT-DETR [53], addresses the above issues and allows for real-time computing. It also leads to SOTA performance and outperforms YOLOv8 on the COCO dataset. Nevertheless, YOLOv9 performs slightly better than RT-DETR, while being trained from scratch (unlike RT-DETR, which is pre-trained on a large object detection dataset).

Common losses – Finally, object detection boils down into: 1) a classification task, with the prediction of the objectness scores and the class labels, and 2) a regression task, with the prediction of the bounding box coordinate offsets. The training thus relies on the use of two types of losses. A commonly used classification loss is the cross-entropy, defined as:

$$L_{\text{CE}} = -\sum_{i=1}^{n} t_i \log(p_i), \tag{2.2}$$

with n the total number of classes, t_i is the i^{th} element of the ground-truth vector $\mathbf{t}$ and p_i the Softmax score (obtained by applying the Softmax activation function on the output digits) for the i^{th} class.

For the regression task, object detectors often employ the Smooth L1 loss,

which is a combination of L1 and L2 loss:

$$L_{\text{SmoothL1}}(x, y) = \begin{cases} 0.5(x-y)^2/\beta, \text{ if } |x-y| < \beta, \\ |x-y| - 0.5 \times \beta, \text{ otherwise,} \end{cases}$$

where x is the predicted value, y the target value, and β a parameter indicating where to switch from L1 to L2 loss. Compared to L2 loss, L1 smooth loss is less sensitive to outliers.

2.1.3 Segmentation networks

Segmentation differs from object detection in that it provides a pixel-level classification. It is generally not used directly for object detection, although in some cases, object-level predictions can be extracted from segmentation maps by using operators (e.g., connected component labelling, possibly along with mathematical morphological filters such as openings and closings) to group pixels into objects. There are two main categories of segmentation: 1) semantic segmentation, which assigns a class label to each pixel in the image, but does not distinguish between different instances of the same class, and 2) instance segmentation, which distinguishes between individual instances of each object class.

Common architectures – Semantic segmentation networks are based on encoder-decoder architectures and reconstruct a segmentation map with the same resolution as the input. Fully Connected Networks (FCN) [54] is a pioneering work in the use of deep learning for semantic segmentation. It consists of replacing the fully connected layers in classification networks (such as AlexNet) by a transposed convolution, which allows for the reconstruction of a full-resolution segmentation map.

At the same time, [5] proposed U-Net, an encoder-decoder architecture with symmetric skip connections, illustrated on Figure 2.7. Unlike FCN, U-Net decoder consists of several upsampling layers. It also introduces skip connections by concatenating features from the encoder branch with the corresponding features (in terms of resolution) in the decoder branch. U-Net has proven to be highly effective for medical and other fine-grained segmentation tasks, and serves as the foundation for many state-of-the-art methods. Variants of U-Net mainly modify the encoder architecture (e.g., using a ResNet backbone), or introduce some attention mechanisms (e.g., MA-Net [55]). Another well-known architecture is DeepLab [56], which improves the segmentation boundaries by introducing atrous convolutions and fully connected Conditional Random Fields.

Finally, instance segmentation can be achieved by combining object detectors and a segmentation head. For example, Mask R-CNN [57] extends Faster R-CNN

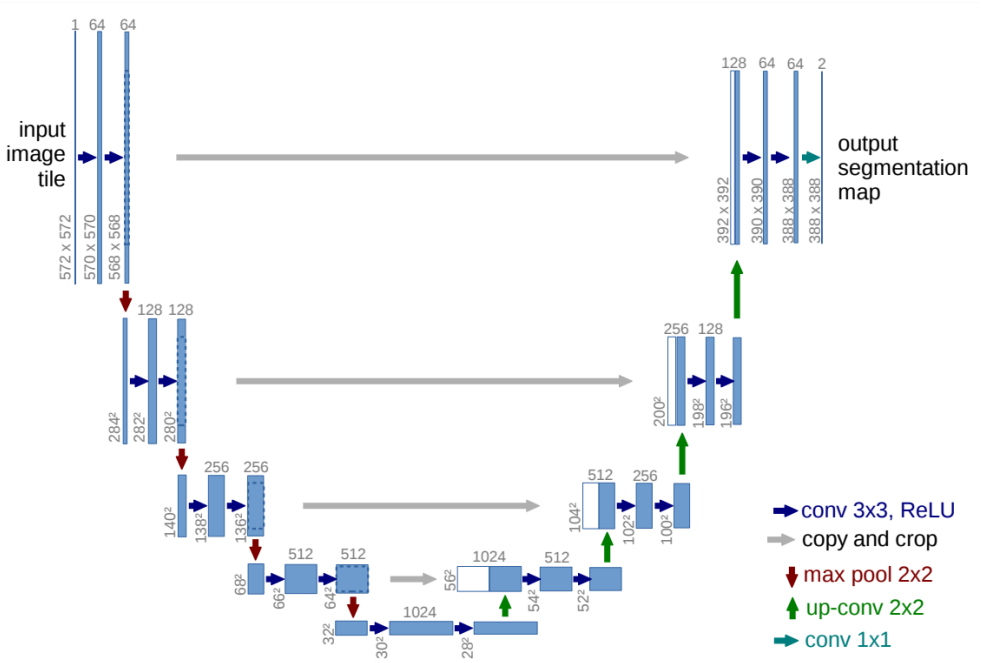


Figure 2.7: Illustration of the U-Net architecture. Figure taken from [5] (Fig. 1). Copyright ©2015 Springer International Publishing Switzerland

bounding-box regression branch with a parallel branch specifically designed for predicting object masks.

Common segmentation losses – Since segmentation boils down to a pixel-level classification task, some segmentation losses are derived from common classification losses. Indeed, the cross-entropy loss can be used as a segmentation loss. However, in contrast to image-level classification, the pixel-level loss is summed over all image pixels, which can be problematic in the case of class imbalance. Indeed, the minority class will be less penalised in case of errors. Therefore, the weighted cross-entropy (WCE) loss was introduced to address this issue. It incorporates a weighting parameter α_i for each class into the cross entropy loss. The lower the number of examples for a class i, the higher the associated α_i value must be set to re-balance the weight of the minority class(es) in the loss. Eq. (2.2) can be reformulated as follows:

$$L_{\text{WCE}} = -\sum_{i=1}^{n} \alpha_i t_i \log(p_i),$$

An improved version of the WCE loss has been introduced, namely the focal loss [27]. Specifically, the focal loss adds a γ parameter to the loss that helps to

focus on hard cases:

$$L_{\text{Focal}} = -\sum_{i=1}^{n} \alpha_i (i - p_i)^{\gamma} \log(p_i),$$

where n is the total number of classes, and p_i the predicted score for class i.

Other segmentation losses have been proposed to directly optimise some segmentation metrics. This is the case of the soft-IoU loss [58]. The latter implements an approximation of the IoU (defined in Section 2.1.4) as follows:

$$L_{\text{Soft-IoU}} = \frac{1}{C} \sum_{c=1}^{C} \left(1 - \frac{\sum_{i=1}^{n} p_{i,c} t_{i,c}}{\sum_{i=1}^{n} (p_{i,c} + t_{i,c} - p_{i,c} t_{i,c})} \right),$$

where C is the total number of classes, $p_{i,c}$ is the predicted value for pixel i for class c, $t_{i,c}$ the ground-truth value (0 or 1) for pixel i and class c. The computation of the IoU is here approximate because the predictions are made in a continuous space (while the IoU is usually computed on binary segmentation maps).

Another widely-used loss is the Dice loss, which is an approximation (still in the sense that it can be applied on non boolean values) of the F1 score. Its computation is similar to the IoU loss, except for the denominator which computes the sum of ground-truth and predicted areas instead of their union:

$$L_{\text{Dice}} = \frac{1}{C} \sum_{c=1}^{C} \left(1 - \frac{2 \sum_{i=1}^{n} p_{i,c} t_{i,c}}{\sum_{i=1}^{n} (p_{i,c} + t_{i,c})} \right).$$

Both Soft-IoU and Dice losses suffer from the following issue: they are equal to zero if the intersection is empty, no matter how far away from each other the ground truth and the predicted detection are spatially. To address this issue, the generalised IoU loss was introduced [59].

2.1.4 Evaluation metrics

Several evaluation metrics are introduced in order to quantitatively evaluate the predictions of an object detector.

Precision, recall and F1-score – The precision and the recall are fundamental and intuitive metrics. Indeed, they respectively indicate how precise the detector is (i.e., does the detector produce a large number of false alarms?) and how many good detections are retrieved by the detector. More specifically, they are defined as follows:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN},$$

where TP stands for True Positive, FP for false positive (i.e., a false alarm) and FN for false negative (i.e., a missed detection). TP, FP and FN can be computed at object or pixel (for segmentation) levels, and a detector is said to be of higher quality when it achieves both a high level of precision and recall. These metrics are complementary, and a summary of the model performance can be obtained through the computation of the F1-score:

$$F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

IoU – For segmentation networks, a commonly used metric is the Intersection over Union (IoU), which is the Jaccard index. It is defined as follows:

$$IoU = \frac{A_{inter}}{A_{union}},$$

where A_{inter} and A_{union} represent the intersection and the union (in numbers of pixels) between the predicted regions and the ground-truth one, respectively. This metric is also important in object detection: indeed, it is used to determine whether an object-level prediction is a true or a false positive. A low IoU will indicate an object-level False Positive.

Average Precision – Object detectors are generally evaluated in terms of mean Average Precision (mAP). mAP computes the area under the precision-recall curve (Average Precision (AP)), averaged over all the object classes. mAP can also be averaged over multiple IoU thresholds to evaluate the network robustness towards localisation errors.

2.2 State-of-the-art methods for infrared small target detection

Small target detection is a real-world application that relies on object detection or segmentation methods to be solved. The specificity of this application lies in the fact that the objects are particularly small: indeed, the Society of Photo-Optical Instrumentation (SPIE) describes a small target as having a spatial area lower than 0.15% of the image, with a low contrast ratio (below 15%) [60]. The methods discussed in the previous section were primarily designed and evaluated on datasets containing medium to large objects, making them suboptimal for detecting small targets.

In the literature, researchers have addressed several challenges inherent to small target detection, such as the information loss occurring during successive downsamplings, the difficulty of distinguishing targets from complex backgrounds, and the imbalance between true detections and false alarms. Proposed solutions include better integration of multi-scale information, or incorporating priors about small targets into the training process.

Given the large number of deep learning-based methods proposed for IRSTD, this section will highlight only the most emblematic approaches. For a comprehensive review of existing methods, readers are referred to the following surveys: [15, 61].

MDvsFA-cGAN – MDvsFA-cGAN [9] is one of the first deep learning-based methods proposed for IRSTD. This framework addresses the tasks of reducing missed detections (MD) and false alarms (FA) by separating them into two distinct subtasks, achieving balance through adversarial training of these models. Specifically, it employs a conditional generative adversarial network (cGAN), which consists of two generators and a discriminator. The generators are used for the generation of a segmentation map, and each of them has a specific role: generator G1 focuses on reducing missed detections, while generator G2 aims to reduce false alarm rate. The discriminator network is then used in adversarial training to balance these tasks. During inference, the segmentation result is obtained by averaging the outputs of the two generators. Although it has demonstrated efficiency, especially when compared to traditional small target detectors, it is computationally expensive and has been surpassed by other deep learning methods.

ACM – In order to better capture finer details such as targets, Asymmetric Contextual Modulation (ACM) has been proposed [6]. Specifically, it modifies the feature fusion in the decoder branch by introducing a top-down channel attention and a bottom-up pixel-level spatial attention mechanisms in an asymmetric way, as illustrated on Figure 2.8. This helps the network to better extract high-level semantics and low-level details. The asymmetric (i.e., spatial attention for one branch, channel attention for the other) as well as the bi-directionnal (both top-down and bottom-up) modulations have been shown to significantly improve small target detection performance. Such a strategy has been applied to both FPN and U-Net architectures.

LSPM – To limit confusion between targets and background noise, [26] introduces Local Similarity Pyramid Modules (LSPMs) in the decoder branch. These modules quantify the degree of similarity between a pixel (or region) and other pixels (or regions, respectively) in the feature map. Pixel-wise similarity is calculated

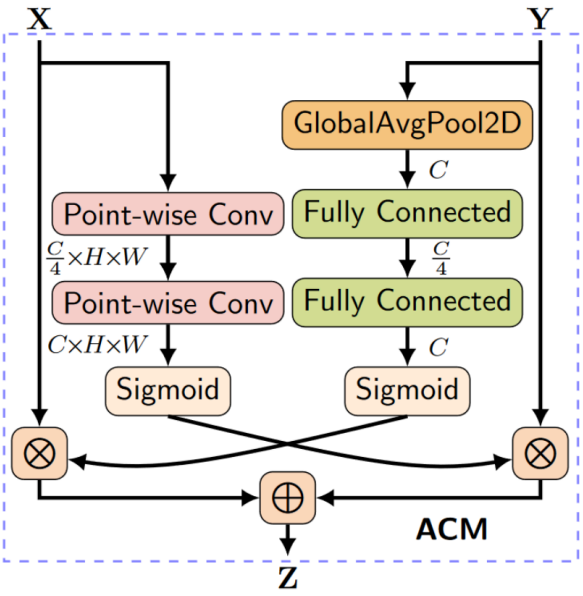


Figure 2.8: Illustration of the asymmetric contextual modulation (ACM). X represents the features extracted by the encoder that serve for the skip connections in U-shaped networks, and Y represents the up-sampled feature maps in the decoder branch. Figure taken from [6] (Fig. 5). Copyright © 2021, IEEE.

through matrix multiplication between the feature map and its transposed version, similarily to the self-attention mechanism discussed in Section 2.1.1. Region-wise similarity scores are obtained by dividing the feature map into a grid. These similarity scores are added to the initial feature map and used as global guidance in the decoder via a channel attention-based feature aggregation module.

ISNet – In order to accurately delineate the shape of IR small targets, [7] introduced IR shape network (ISNet), illustrated on Figure 2.9. The latter has a U-Net structure, and is trained with two objective functions, namely a segmentation loss (e.g., Dice loss), and a specific edge loss. To obtain the edge supervision, the input is passed through a Sobel filter, and then through several Taylor fine difference (TFD) blocks that enhance the edge information and improve the contrast between the target and the background by aggregating edge details from different levels. [7] also introduces two orientation attention aggregation (TOAA) blocks in the decoder. These aim to better capture target shapes and suppress high-frequency noise, by computing attention maps along one direction (row or column) using deformable convolutions.

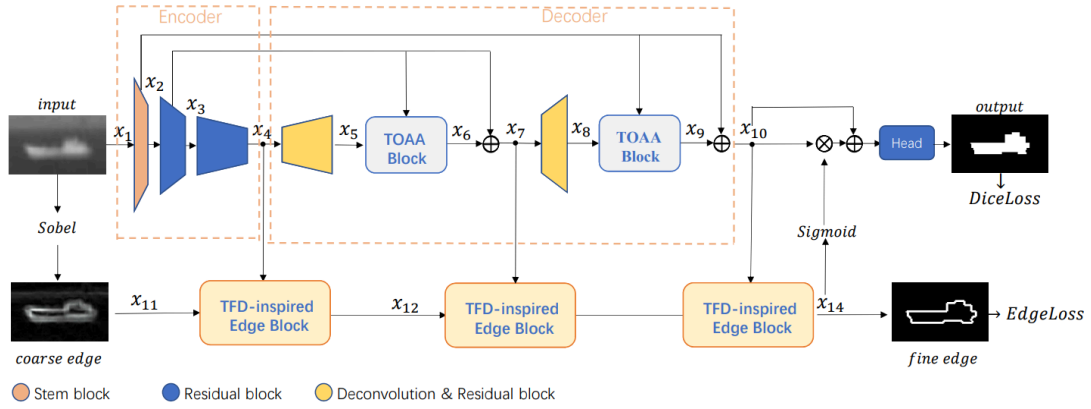


Figure 2.9: Illustration of the ISNet network. Figure taken from [7] (Fig. 1). Copyright ©2022, IEEE.

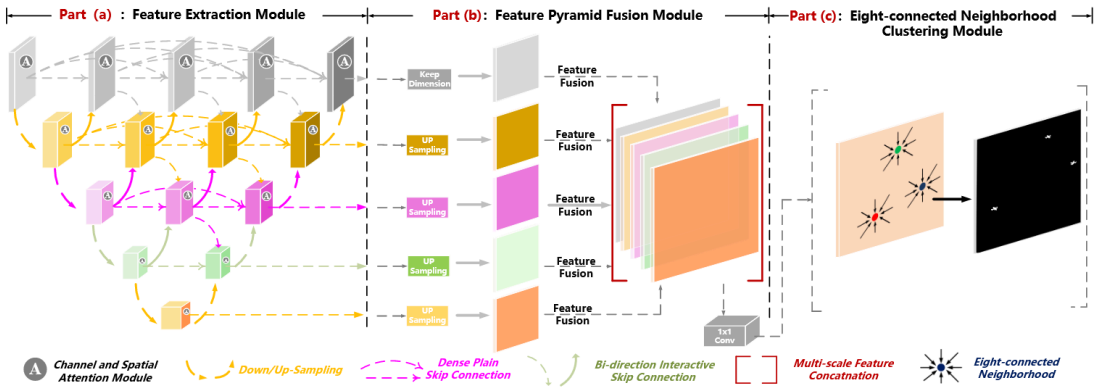


Figure 2.10: Illustration of the DNANet architecture. Specifically, DNANet is composed of a dense-nested U-shaped backbone (DNIM), and a Feature Pyramid Fusion Module (FPFM). Figure taken from [8] (Fig. 3). Copyright ©2022, IEEE.

DNANet – DNANet [8] proposes to limit the information loss on small targets due to pooling layers by introducing 1) a dense-nested U-shaped backbone (DNIM), and 2) a pyramid fusion module (FPFM). Specifically, the DNIM backbone consists of several U-shaped subnetworks with channel and spatial attention modules that are stacked together and densely connected, as illustrated on Figure 2.10. Such a process has been shown to benefit the representation of small targets in deeper layers, and has been used in other works (e.g., [62]). In order to better aggregate the multi-scale feature maps, the FPFM block takes as input the last feature map of each scale, upsamples them to the full-resolution, concatenates them and applies a 1×1 convolution to obtain the final segmentation map. DNANet has shown impressive performance on widely used small target detection datasets and is considered as one of the SOTA methods for IRSTD. An extension of DNANet using a Swin Transformer backbone has also been proposed in [63] (IDNANet), which further improves the performance on several IRSTD benchmarks. This encourages the future development of IRSTD methods based on ViT encoders.

AGPCNet – Concurrently, AGPCNet [64] proposes to integrate attention mechanisms in the deep layers and an asymmetric fusion module (AFM, slightly different from the one proposed in ACM) to increase both the receptive field and the feature representation capability of the network. More specifically, AGPCNet divides the deep feature maps into a grid and computes 1) local attention scores for each patch, which indicate the probability that each pixel is a target within the given patch, and 2) global attention scores (i.e., patch-level attention), which indicate for each patch the probability of containing a target. In terms of the feature attention module, AFM differs from ACM in that it multiplies the sum of deep and low-level feature maps by both channel and pixel attention maps. This strategy has been shown to improve overall performance compared to ACM.

MTU-Net – MTU-Net [65] proposes a multi-level TransUNet [42] and a new loss, namely the FocalIoU loss. Specifically, the network architecture consists in computing long-range dependencies through a ViT module for each features output by the different levels of the encoder. These features are then merged together (concatenation and convolution) and passed through a conventional U-shaped decoder with skip connections. Regarding the loss, FocalIoU combines both Focal and Soft-IoU losses as follows:

$$L_{\text{FocalIoU}} = 2(1 - L_{\text{Soft-IoU}}) L_{\text{Focal}}^{\frac{1+L_{\text{Soft-IoU}}}{2}}.$$

MTU-Net has lead to impressive performance on the challenging ship detection task compared to SOTA infrared small target detectors such as DNANet.

Method	F1
MDvsFA-cGAN	69.8
ACM (U-Net)	77.6
LSPM	76.2
DNANet	**85.9**
AGPCNet	<u>84.5</u>

Table 2.2: Performance obtained by some SOTA methods on SIRST dataset. The pixel-level F1 scores are provided by [15]. The best result is given in bold, and the second best result is underlined.

The survey paper [15] proposes to train and evaluate several SOTA methods for IRSTD on different datasets. The methods evaluated include MDvsFA-cGAN, ACM (U-Net version), LSPM, DNANet and AGPCNet. Table 2.2 shows the pixel-level F1 scores achieved by these methods on the SIRST dataset (presented in section 2.3.2). The results are taken from [15] (Tab. 6). It can be seen that DNANet performs best, closely followed by AGPCNet. MDvsFA-cGAN gives poor results compared to the other methods. Therefore, we will consider DNANet, AGPCNet, LSPM and MTU-Net (which is a newer architecture) as our SOTA baselines in the remainder of the manuscript.

2.3 Small target detection datasets

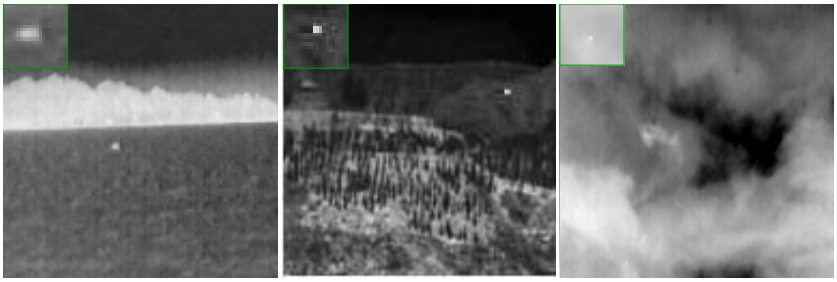
Let us present some single-frame datasets for small target detection, which will later be used to train and evaluate several deep learning methods for small target detection. We will consider five datasets. First, we consider MFIRST [9], SIRST [6] and IRSTD-1k [7], which all tackle infrared small target detection. Note that since the objects contained in these datasets are particularly small and of low resolution, they are all grouped under one class category, namely the "target" class. We also consider two other datasets that will allow us to evaluate the generalisation ability of our methods to other applications, namely VEDAI [10] (vehicle detection from remote sensing imagery) and S2SHIPS [11] (ship detection from remote sensing data). The main characteristics of each dataset are presented in Table 2.3, and we specify them in the following paragraphs.

2.3.1 MFIRST

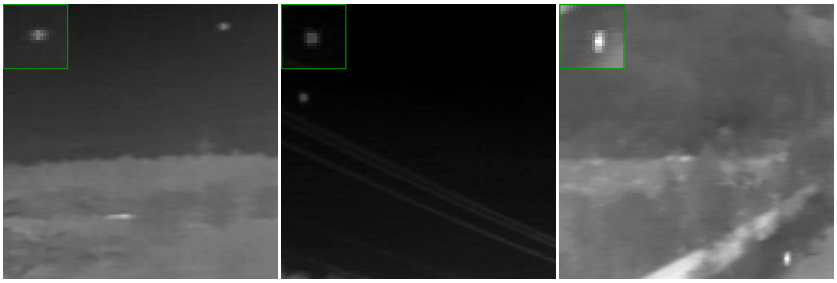
The first large-scale IRSTD dataset proposed in the literature is MFIRST [9], which consists of 10100 real or simulated IR images of various targets evolving on complex backgrounds. This dataset has enabled the development of deep learning

Dataset name	Sensor	Image size	#img.	Description
MFIRST [9]	IR	from 173×98 to 407×305	10k	Infrared small targets evolving on different backgrounds, including clouds, buildings, and vegetation. Most targets are simulated using crops of real targets or a two-dimensional Gaussian function. Real targets include cars, drones, birds, cats, planes, etc.
SIRST [6]	IR (NIR, SWIR, MWIR)	from 135×96 to 456×278	427	Infrared small targets evolving on different backgrounds, including textured clouds, buildings, and vegetation. Targets mainly include vehicles and aircrafts.
IRDST-1k [7]	IR	512×512	1000	Infrared small targets evolving on different backgrounds, including the sea surface, fields, mountain areas, urban areas or clouds. Targets include drones, creatures, vessels and vehicles.
VEDAI [10]	RGB and IR (NIR)	1024×1024	1200	Small vehicle detection from remote sensing data. The dataset contains nine classes of vehicles, including plane, boat, car, etc. The objects evolve on different backgrounds such as fields, grass, mountains or urban areas.
S2SHIPS [11]	Multispectral (443nm – 2.19µm)	1783×938	16	Small ship segmentation from multispectral satellite imagery (Sentinel 2 sensor). The images present challenging conditions, such as coastlines, cloud cover or rough sea.

Table 2.3: Datasets considered for small target detection and their characteristics.



a) Real targets



b) Simulated targets

Figure 2.11: Example of images taken from MFIRST dataset [9]. Both a) real and b) simulated targets are displayed. Targets are enlarged in the top left corner.

methods for detecting, for example, drones or vehicles in urban or wild areas. Some examples are shown in Figure 2.11, with a distinction between a) real and b) simulated targets. MFIRST is composed of 100 real IR images that are extracted from 11 IR sequences, and is augmented with 10000 simulated images. For this purpose, background images are collected from the Internet, and then small targets are overlaid. These targets are either extracted from real IR images (e.g., the last image in Figure 2.11b)), or simulated by a two dimensional Gaussian function (e.g., the first two images in Figure 2.11b)). The original paper proposes to train on simulated data and to keep real data for evaluation only. Although this dataset contains a large number of examples and can be used to develop deep learning methods, the fact that it is mostly composed of simulated images (i.e., with IR signatures that are far from the reality) makes it difficult to consider its use as a benchmark dataset. For this reason, there will be very limited mention of it in the remainder of the manuscript.

2.3.2 SIRST

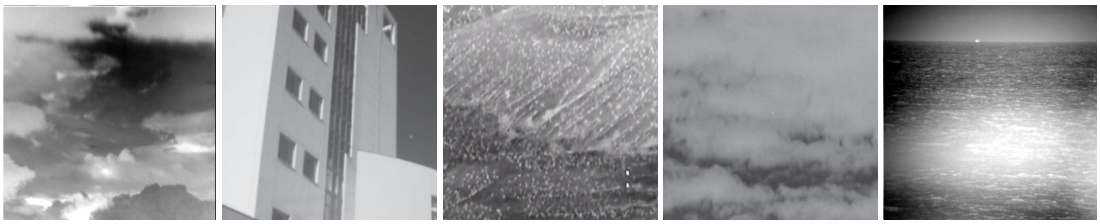


Figure 2.12: Some examples of images from SIRST dataset [6]. There are six small targets hidden in these images, can you spot them all?

SIRST [6] dataset, also referred to as NUAA-SIRST, is one of the first publicly released real-image infrared small target datasets, and it is widely used in the literature as a reference dataset for IRSTD. This dataset contains 427 real monospectral infrared images (in NIR, SWIR or MWIR domains), with resolution ranging from 135×96 to 456×278. Some examples are shown in Figure 2.12. 90% of the images contain a single target, and most targets follow the definition of a small target proposed by the SPIE, i.e. objects having a total spatial extent of less than 80 pixels (9×9) [60]. More specifically, 55% of the targets occupy less than 0.02% of the image area.

2.3.3 IRSTD-1k

In this thesis, we also consider a recently published dataset for small target detection, namely IRSTD-1k [7]. This dataset is larger than SIRST (1000 images)

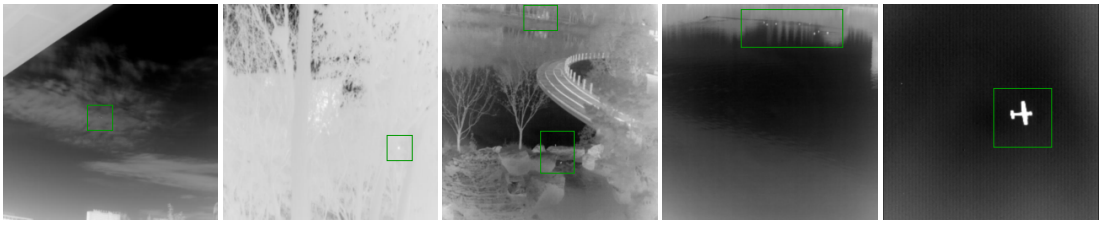


Figure 2.13: Examples of images extracted from the IRSTD-1k dataset [7]. Areas containing targets are framed by a green rectangle.

and contains more challenging scenes, with different kinds of small objects (e.g., aircrafts, animals), as illustrated in Figure 2.13. It also contains some relatively large objects, as shown in the last image of Figure 2.13. Since our work focuses on developing and evaluating methods for *small* target detection, we decide to remove the images that contain targets having a spatial extent larger than 90 pixels (this represents 15% of the dataset), and refer to the filtered dataset as "IRSTD-850".

2.3.4 VEDAI

Figure 2.14: Example of IR images taken from the VEDAI dataset [10]. Vehicles are framed in green.

In order to assess the generalisation capabilities of the developed methods to other applications that are slightly different from IRSTD, we propose to consider the VEDAI dataset [10]. The latter is used for benchmarking vehicle detection in aerial images, and is composed of 1200 RGB images and their associated infrared images, both of size 1024×1024. It contains nine different classes of vehicles (including, for example, plane, boat, car or truck), and the objects evolve on various backgrounds such as fields, mountains or urban areas. In contrast to MFIRST, SIRST or IRSTD-1k, VEDAI contains objects of medium size that have better spatial resolution (so the objects have more texture) and are more numerous in an image. Indeed, there is an average of 5.5 vehicles per image, and they represent about 0.7% of the image area. Some examples are shown in Figure 2.14 (IR image only). One great challenge of this dataset is that there are many background

elements, such as buildings or other infrastructures, that can lead to false alarms.

2.3.5 S2SHIPS

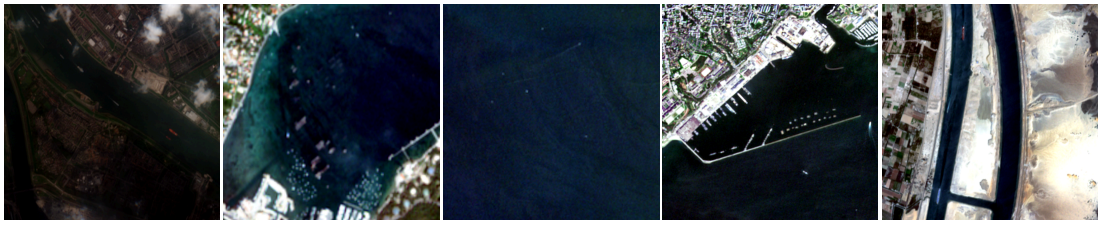


Figure 2.15: Examples of image patches extracted from S2SHIPS dataset [11].

Another challenging small object detection dataset we considered is the S2SHIPS [11] dataset, which tackles the segmentation of small ships from remote sensing data. S2SHIPS dataset is composed of 16 multispectral images taken by the Sentinel2 satellite sensor, each of size 1783×938 pixels. The images include RGB, NIR and SWIR bands. Some examples are shown in Figure 2.15 (RGB reconstruction). Note that this task is particularly challenging for several reasons: i) there are a large number of tiny ships (e.g., pleasure boats, see the second image in Figure 2.15), ii) the moored or tiny ships are very difficult to distinguish from decks or water wings, iii) the low resolution of satellite data requires the use of subpixel information, and iv) the cloud cover (e.g., as in the first picture of Figure 2.15) makes it very difficult to detect ships, and can even lead to many false alarms.

2.4 Discussion and conclusion

In this chapter, we have first presented the key concepts of deep learning-based object detection. We have introduced feature extraction and the two main networks allowing for this step, namely convolutional networks and Vision Transformers. We have also introduced some famous object detectors and segmentation networks. Although they have led to impressive performance on challenging datasets (e.g., the COCO dataset [35]), SOTA methods lead to weak performance when detecting small/tiny objects. As explained in [66], this is mainly due to the loss of small object information (induced by successive downsamplings), the noisy feature representation (because of the background) or the low tolerance for bounding box localisation errors.

In the literature for IRSTD, several methods have been proposed to address some of the above issues. For example, dense nested U-shaped architectures and specific multi-scale fusion modules have been introduced, which effectively limit

the information loss on small targets and lead to good performance on several IRSTD benchmarks. Some methods also include local and large-scale attention mechanisms in order to limit the confusion between targets and background elements. It can be noticed that SOTA IRSTD methods all rely on segmentation networks. Indeed, object detectors are particularly weak when it comes to small object detection. Very few studies have focused on adapting such detectors for IRSTD [32, 67, 68], and no rigorous comparison was made with SOTA IRSTD methods. Indeed, SOTA methods rely primarily on pixel-level metrics, such as the F1 score, the IoU, the probability of detection (also known as the recall), or the probability of false alarms (which corresponds to the number of pixel false alarms divided by the total number of pixels). However, this means that any good detections or missed detections at the object level cannot be counted. It is possible to extract object-level metrics (such as object-level F1 score or mAP) from segmentation maps by relying on morphological operators, as done in [8]. In this case, another issue occurs. In order to mark an object-level prediction as a true positive, it is common to compute the IoU between the prediction and the ground-truth, and if the IoU is above a certain threshold (often 50%), then the prediction is labelled as a true positive. However, [66] explains that small objects have a very low tolerance to bounding box localisation errors, since a small deviation in the number of pixels induces a significant drop in the IoU with the ground-truth. One solution that has been proposed by several papers taken from the literature of small object detection is to model predicted bounding boxes as 2D Gaussian distributions and evaluate the distance between these distributions (e.g., normalised Wasserstein distance) rather than the IoU between the predicted boxes. [68] effectively relies on this strategy to assess infrared small target predictions.

Another issue that has not been fully addressed in the literature for IRSTD is the severe class imbalance induced by the scarcity of data and of small target samples. Indeed, segmentation networks particularly struggle in learning from class-imbalanced datasets. It is possible to overcome this issue by using an appropriate loss function (e.g., weighted cross-entropy or focal loss), or by artificially increasing the number of target samples (e.g., using super-resolution). However, these strategies do not take advantage of the *unexpectedness* of small objects with respect to the background, as one could do in an anomaly detection approach with, for example, one-class classifiers [29], that discriminate small objects as *unexpected* patterns with respect to the background. Such a criterion can efficiently reduce the number of false alarms induced by the background and thus can allow for a better balance between precision and detection rate.

In the following, we will propose several approaches to improve the detection of small targets, which we will divide into two parts. In the first part, we will introduce a new learning paradigm specifically dedicated to small target detection.

The latter is based on the *a contrario* theory, and is inspired by anomaly detection methods. This allows us to introduce an *a priori* on small targets (in fact, small targets are *unexpected*) and to control the number of false alarms. In the second part, we will look at unsupervised learning methods, in particular self-supervised learning methods, to help and induce more robustness in the feature extraction of small objects. More specifically, we will ask to what extent and under what conditions these methods can be beneficial for IRSTD.

Part I

A contrario paradigm for infrared small target detection

Chapter 3

A contrario formulation and small object detection

This chapter is the first in the part entitled "*A contrario* paradigm for infrared small target detection", which constitutes the first part of this manuscript. This part is divided into 3 chapters. The aim of this first chapter is twofold: firstly to introduce the *a contrario* theory and the key concepts on which it is based, and secondly to propose a "proof of concept", demonstrating the benefits of using this theory to detect small targets. This will justify the contributions proposed in the next two chapters, namely the integration of an *a contrario* criterion into the training loop of a segmentation network, then its integration within a detection network.

In this chapter, we will first present our intuition for the proposition of using the *a contrario* theory for IRSTD. After that, we will briefly present the theories of perception, which are at the foundation of the *a contrario* methods, and then introduce the *a contrario* formulation in a general framework. Finally, we will conclude with an experimental section in which we will propose two *a contrario* post-processing steps to filter final segmentation maps, and highlight their benefits for our application.

3.1 Intuition

In order to identify potential reasons or mechanisms giving rise to false alarms or missed detections, we first study qualitatively how the feature maps of a small target detection network are learned. As presented in the previous chapter, segmentation neural networks are at the state-of-the-art for small target detection based on deep learning. We therefore analyse the training behaviour of conventional segmentation baselines, especially in the case of frugal training. Indeed, a

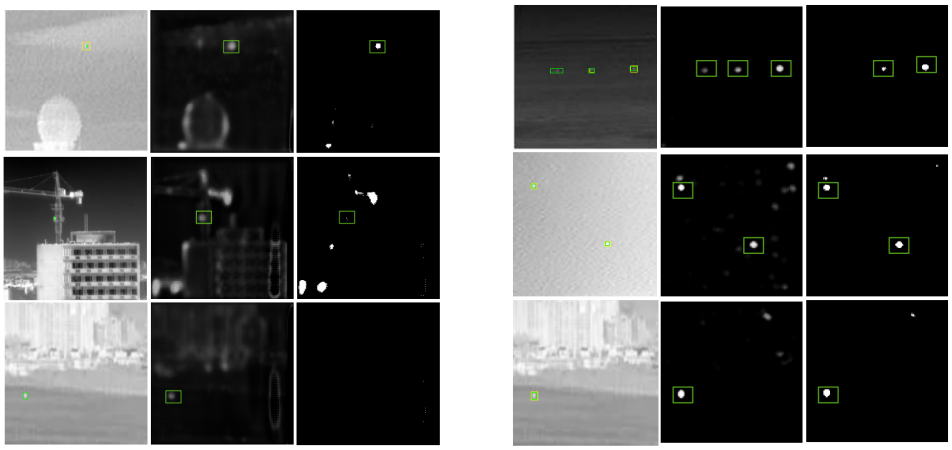
drawback of segmentation networks is that they are strongly affected by class imbalance, and unfortunately, in the case of small target detection, the datasets are particularly imbalanced (in terms of number of targets as well as number of pixels per target). Looking at the feature maps under such challenging conditions allows us to identify the underlying mechanisms of deep learning-based small target detection, as well as the strengths and weaknesses of conventional detectors. To do so, we select the conventional ResUnet architecture with a ResNet-18 encoder. We consider the MFIRST dataset, with images resized to a resolution of 256×256, and design two settings: 1) train in a data-sufficient regime (2500 images), 2) train in a frugal setting with only 100 images, which constitutes an important constraint of our application.

Figure 3.1 shows that the score maps extracted by the neural networks are different depending on whether we consider the U-Net trained on sufficient data (Figure 3.1b) or on little data (Figure 3.1a). Indeed, frugal training results in noisier maps. The more robustly the network is trained, the less noisy the feature maps will be. To eliminate this noise, it is common to apply a fixed threshold on the feature maps. By doing so, we assume that the value of the noise observed on background pixels is lower than the pixel intensity of the potential targets. However, we can see on Figure 3.1 the limits of such hypothesis since several false alarms are raised. One solution would be to increase the value of the threshold, but this can also lead to missed detections. The threshold choice seems to be crucial and yet challenging, and the gray level value itself may not be the only criterion to take into account. A more specific way of thresholding should be designed for taking into account, for example, the density or the shape of the hot spots, or the *unexpectedness* of the target features in contradiction with the background features. These criteria can be formalized by taking into account the perception patterns that are specific to small target detection. In the following, we introduce a theory that models the perception laws in a very general way, namely the *a contrario* paradigm.

3.2 Perception theory

3.2.1 Vision science and optical illusion

To understand the laws and theories of perception introduced in the next paragraphs, let us first recall some important historical stages in the science of vision. The first studies were aimed at explaining how vision works from a biological and physical point of view. They focused mainly on the anatomy of the eye (e.g., the principle of peripheral and foveal vision introduced by da Vinci) and the path of light rays. These studies include the theories of emission supported by Plato or



(a) Detector trained with 100 images. (b) Detector trained with 2500 images.

Figure 3.1: Examples of detector predictions on 3 images. From left to right: original image with ground truth framed in green, score map at network output (ground truth in green), prediction after thresholding with fixed threshold where good detections are framed in green. All the white areas on the prediction maps that are not framed in green are false alarms.

Aristotle, who believe that visual perception is produced by light rays emitted by the eyes. At the same time, the theory of intromission is emerging, stating that visual perception is produced by the reflected "image" of objects. This theory was supported by Democrite and Epicurius among others, and Alhazen validated it experimentally, proving the laws of reflection and refraction. The latest theory was later modernized and refined, with Newton's corpuscular theory and wave theory among others.

Nevertheless, the brain plays an important role in the perception of the world. Helmholtz [69] was one of the first to propose a modern study of visual perception and to introduce the unconscious inference. This term describes an involuntary and unconscious mechanism that allows us to form visual impressions in our brain. Visual aberrations or illusions are evidence of the fact that our brain plays a major role in the perception of our environment.

Let us take the example of geometric–optical illusions. These are visual illusions in which the geometric properties of what is seen differ from those of the corresponding objects. Helmholtz illustrates this phenomenon with Müller-Lyer illusion, presented in Figure 3.2. It consists of drawing three lines with different endings. Depending on the endings (which constitutes the distorting element), we may think that one line is longer than the others, but in reality, they are all of the same length. The lines are said to be the distorted elements. The fact that we can see something that is not really there, or that we can deform objects, is an

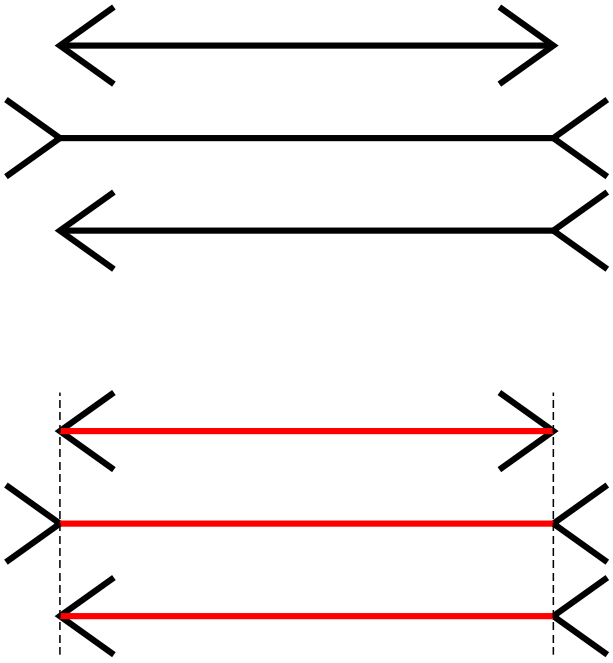


Figure 3.2: Müller-Lyer illusion. Copyrights Fibonacci, CC BY-SA 3.0 via Wikimedia Commons.

indication that our perception is more than an exact copy of reality. When only part of an image, such as when one object hides another object, is presented to the brain, our visual perception constructs a complete impression of the scene. This constructive human perception allows us to have a comprehensive understanding of our environment, even when information is limited, and it also allows us to create meaningful context. This is possible because of some assumptions we make about our world, based on our past experiences. Some examples of such assumptions are that light comes from above, faces are upright, objects tend to have convex borders, etc. Few theories have attempted to explain this phenomenon theoretically. In the following, we present one of them, namely the Gestalt psychology.

3.2.2 Gestalt theory

One question about visual perception is how do we go from an innumerable set of isolated singular elements to the formation of objects. The Gestalt theory attemps to provide an answer by identifying how we can perceive a structure. This theory was founded by three German psychologists, namely Kurt Koffka [70], Wolfgang Kohler, and Max Wertheimer [71], in the early 20th century. Koffka suggested that "the whole is more than the sum of its parts", and, more specifically, that "the whole-part relationship is meaningful". He explained that our brain has an

innate ability to analyse a scene and perceive a unification of elements rather than individual objects. The Gestalt psychology tries to model this phenomenon by introducing several grouping laws. Indeed, some dots may have some characteristics in common, and thus they can be joined together to form a larger visual object, namely a gestalt. Figure 3.3 illustrates some of the laws that determine how the visual system automatically groups elements into patterns:

- Colour constancy: Regions with similar colour can be grouped together. For example, in Figure 3.3a, we see the black spot as a whole, rather than several black dots.
- Vicinity: close elements (compared to the surrounding elements) can be unified (Figure 3.3b).
- Similarity: Similar groups of objects (patterns) are grouped together. They therefore form a higher level object (Figure 3.3c).
- Closure: a closed curve can define an object (inside the curve) and a background (outside the curve) (Figure 3.3d).
- Constant width: Two parallel curves can describe an object (Figure 3.3e).
- Good continuation: an alignment of dots or objects can define a contour (Figure 3.3f).
- Symmetry: we can group objects together if they form a symmetry (Figure 3.3g).
- Amodal completion: a T-junction (i.e., when a curve stops another curve) can suggest that an object is occluded by another. However, this suggestion, which is based on the good continuation law, can lead to several different interpretations, as shown on Figure 3.3h.

Additionally, it should be noted that all grouping gestalt laws are recursively applicable. They can be applied initially to atomic inputs before being applied again in the same way to partial gestalts that have already been constituted.

3.2.3 Helmholtz principle

The gestalt laws are stated as independent grouping laws that start from the same building elements. Thus, conflicts between grouping laws can occur, which can lead to conflicts between different interpretations. These different interpretations may lead to the perception of different and sometimes incompatible patterns in a

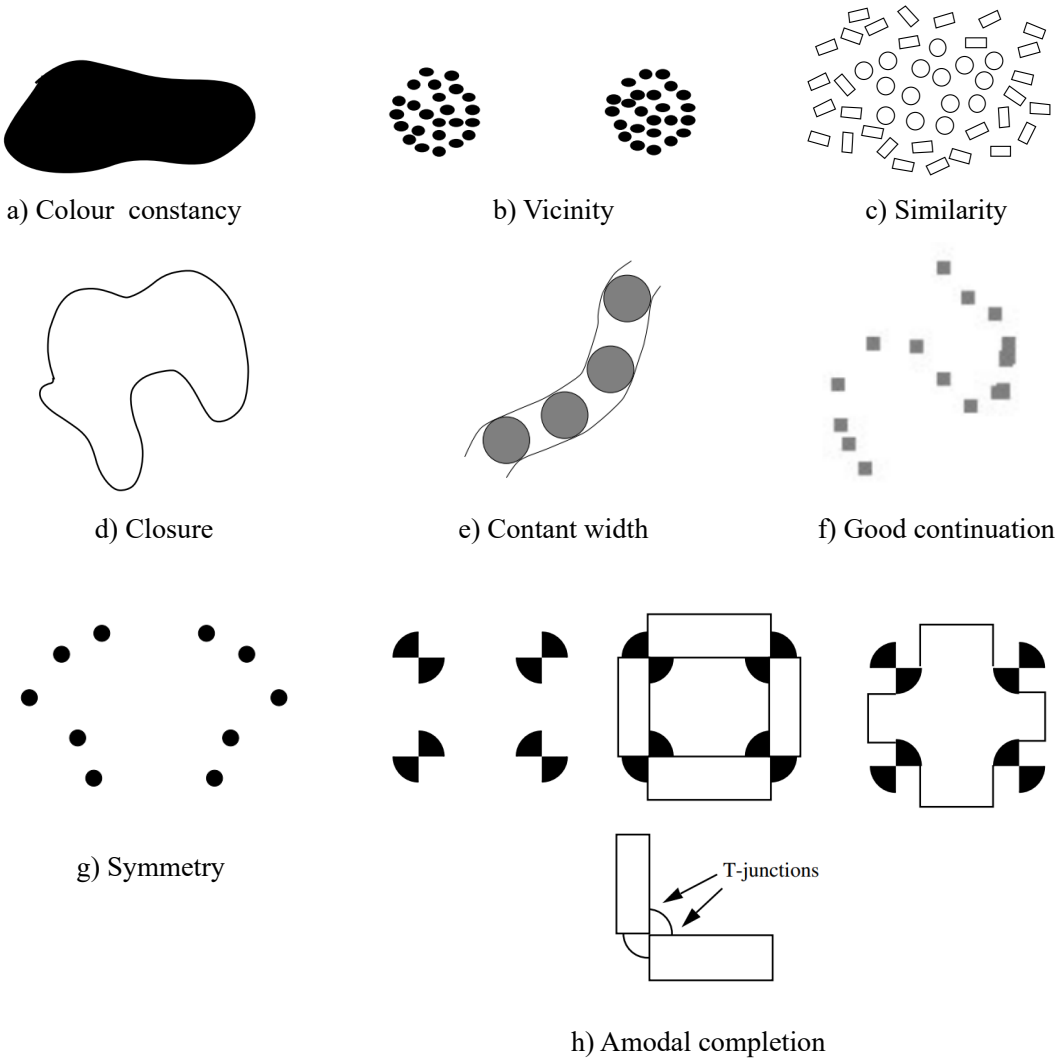


Figure 3.3: Some fundamental Gestalt laws. Sub-figures are taken from [12].

given figure. This is, for example, the case with Figure 3.4: depending on which part you look at first, you may see a rabbit, while others may see a duck. However, it is impossible to see both animals at the same time.

There are several types of conflicts, but we will focus on the masking by texture as it is the one we encounter the most often in our work. Masking occurs when partial gestalts are hidden by other partial gestalts. This can be illustrated with Figure 3.5. On Figure (b) we can perceive an alignment of four segments, while this is not the case in Figure 3.5a (which nevertheless contains the four segments from Figure 3.5b).

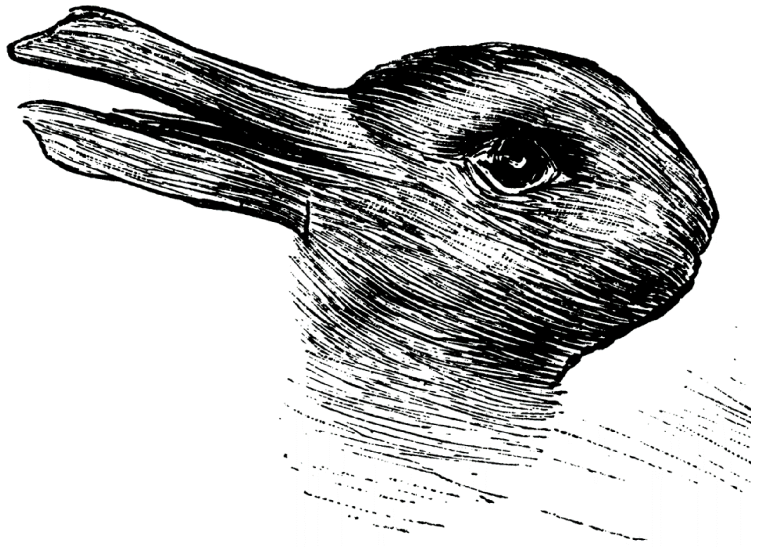
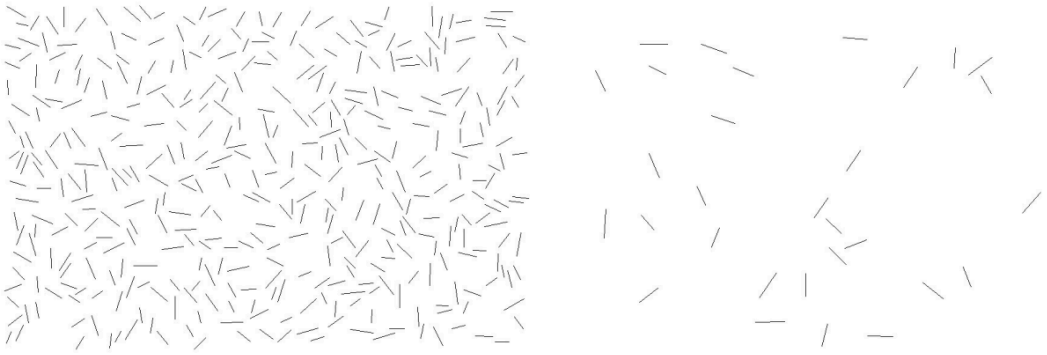


Figure 3.4: The famous duck-rabbit optical illusion.

The multiplicity of segments in Figure 3.5 induces a masking by texture, so that we cannot perceive the alignment of some of them. This suggests that a grouping law can only be active in an image if its application would not create a huge number of partial gestalts. This principle of non-accidentalness is referred to the Helmholtz principle which states that a structure is perceived when a significant deviation from randomness occurs. As a result, we can divide image objects or relationships between objects into two categories: those that occur by chance and those that are the result of a meaningful structure. Such a theory is the basis of the *a contrario* paradigm, which we describe in the next section.



(a) (b)

Figure 3.5: Illustration of the Helmholtz principle. The figures are taken from [12].

3.3 *A contrario* formulation

The *a contrario* paradigm is a statistical approach for detecting geometric structures in an image that is inspired by the Helmholtz principle. It consists in estimating the *significance* of a possible structure in contradiction to a random model. This model, called the null hypothesis H_0, will be referred to here as the "naive" model, and will represent the idea of what is "normal" background, i.e. without any structure or object. One of the benefits of such an approach is that it does not rely on any prior knowledge about the image structures: it only assumes the naive model representing the background noise. Even further, the assumptions made on the background can be very approximate. Indeed, in *a contrario* reasoning, the "naive" hypothesis made on the background distribution only needs to be contradicted in the presence of any structure of object of interest. It means that the background distribution modelling can be only approximate, as long as the objects of interest fall outside this distribution. Moreover, conversely to the targets, for the background, we have a great amount of information at our disposal. The *a contrario* paradigm has been applied on several use cases such as the detection of meaningful segments or stripes [12], or breast cancer detection [72]. The following introduces the mathematical formulation of the *a contrario* paradigm.

Multiple testing - The *a contrario* formulation derives from statistical testing, more specifically multiple hypothesis testing formulated as follows. Let us consider n geometric objects $O_1, ..., O_n$ within an image. Let $X = (X_1, ..., X_n)$ be a set of random variables where, for $i \in [\![1, n]\!]$, X_i describes the features of object O_i. These features can be for example the colour of the object, an angle, etc. We define our assumption H_0 as follows: $X_1, ..., X_n$ are independent and identically

distributed (iid) random variables following the naive model. Let $A = [\![1, n]\!]$, $P(A)$ the powerset of A, and $G \in P(A)$, a subset of indices representing a subset of objects. For example, if the objects O_i are the pixels of an image, as in the basic case, we may want to test 2D boxes or clusters of spatially close pixels, i.e. the tested groups G are chosen to be interpretable with respect to the application. The question we want to answer is: are the features $\{X_k\}_{k\in G}$ *unexpected* enough under the assumption H_0 to group $\{O_k\}_{k\in G}$ together? In other words, does the group $\{O_k\}_{k\in G}$ represent a structure?

To answer this question, we can perform a statistical test in order to reject (or not) H_0. Rejecting H_0 proves, *a contrario*, that a structure (gestalt) exists. Let $\eta_{test} \in \mathbb{N}^*$ be the total number of tested groups. This number can represent, for example, the total number of pixels within an image if the pixels are tested independently (one group per pixel), or the number of stripes (defined by a point and an angle for example) if we look for point alignments, that we plan to test. Let $\theta \in \mathbb{R}^*$ be the rejection threshold, F the testing function and G_l be the tested group for each $l \in [\![1, \eta_{test}]\!]$. For the tested group $G_l \in P(A)$, H_0 is rejected if $F(\{X_k\}_{k\in G_l}) < \theta$. In our case, the multiplicity of the statistical test comes from the fact that we are testing different groups G_l. In a more general context, one can also consider several test functions F_l, with their respective rejection thresholds θ_l.

Let $\alpha \in \mathbb{R}^*$ be the maximum value we set for type I error, i.e. the probability of falsely rejecting H_0. In the literature, it is common to control the family-wise error rate (FWER), i.e. the probability of having at least one false alarm. The FWER with a confidence α is defined as follows:

$$\mathbb{P}_{H_0}(\exists l \in [1, \eta_{test}], F(\{X_k\}_{k\in G_l}) < \theta) < \alpha \tag{3.1}$$

When the number of statistical tests that are carried out independently increases, the overall risk of type I error increases. Indeed, repeating the risk of obtaining a significant result by chance on at least one test increases the overall risk of wrongly rejecting H_0. Basically, if each individual test is performed with a risk α, the overall risk after η_{test} tests is $1 - (1 - \alpha)^{\eta_{test}}$. To address this issue, we can apply the correction of Bonferroni. It consists in testing each individual group G_l at a significance level of $\alpha_l = \frac{\alpha}{\eta_{test}}$. This corrected α_l value comes from the Taylor series approximation $(1 - \alpha)^{\eta_{test}} \approx 1 - \eta_{test}\alpha$ for α close to 0, so that $1 - (1 - \alpha)^{\eta_{test}} \approx \eta_{test}\alpha$. Thus, imposing for each test a lower risk $\frac{\alpha}{\eta_{test}}$ allows one to satisfy Eq. (3.1).

However, such a constraint is highly conservative: the probability of detecting an event is particularly low, which means that the control of the type I error, i.e. false alarm in a detection problem, is at the expense of an increase of the type II error, i.e. missed detection in a detection problem. Several procedures that verify Eq. (3.1) and that are less conservative were proposed in the litera-

ture. They focus on having a set of significant tests, and do not constraint each test independently. We can cite for example the iterative algorithms step-down and step-up [73, 74]. However, these methods are more complex, which is not recommendable for integration into deep learning frameworks.

Finally, we emphasize that statistical control (i.e., constraining a probability) may not be sufficient, especially in defense applications where the risk must be tightly controlled in *number* of false alarm, e.g. zero false alarm (when it comes to lethal consequences). In such cases, it seems highly preferable to handle the Number of False Alarms directly. Besides, as shown by [30, 12] and as we will see in the next paragraphs, it allows for a nice computational framework.

Number of False Alarms Let us now introduce the NFA in a more formal way, as well as some useful properties.

Definition 3.3.1 (Number of False Alarms)**.** With previous notations, F defines a NFA provided that, $\forall \epsilon > 0$, and for $X_i \sim H_0$, it is ϵ-meaningful, i.e. the following condition is verified:

$$\mathbb{E}[\#\{l, F(\{X_k\}_{k\in G_l}) \leq \epsilon\}] \leq \epsilon, \tag{3.2}$$

where the symbol $\mathbb{E}[.]$ stands for the mathematical expectation and $\#\{.\}$ for the cardinality of a set.

This property guarantees that, on average, raising a detection every time F is lower than ϵ leads to at most ϵ false alarms allowing thus for the control of the number of false alarms. Then, an observation $\{x_k\}_{k\in G_l}$ is said to be "ϵ-meaningful" if $F(\{x_k\}_{k\in G_l}) \leq \epsilon$, where ϵ is the predefined maximum value for the expected number of false alarms. Thus, the lower $F(\{x_k\}_{k\in G_l})$ is, the more meaningful the detected structure is.

Let us introduce μ a measure. Given an observation x_k, the NFA (i.e., the function F_l in Eq. (3.1)) is often defined as:

$$\text{NFA}(\{x_k\}_{k\in G_l}) = \eta_{test} \times \mathbb{P}_{H_0}(\mu(\{X_k\}_{k\in G_l}) \geq \mu(\{x_k\}_{k\in G_l})), \tag{3.3}$$

Grosjean and Moisan [72] proved that such function satisfies Definition 3.3.1. More generally, they proved the following property:

Property 3.3.1 (Condition on the number of tests)**.** With previous notations, $F(\{x_k\}_{k\in G_l}) = \eta_l \times \mathbb{P}_{H_0}(\mu(\{X_k\}_{k\in G_l}) \geq \mu(\{x_k\}_{k\in G_l}))$, with $(\eta_l)_{1\leq l\leq \eta_{test}}$ a set of positive real numbers, is a NFA provided that $\sum_{l=1}^{\eta_{test}} \frac{1}{\eta_l} \leq 1$.

In practice, we fix $\eta_l = \eta_{test}$, which satisfies the Property.

In summary, the NFA formulation can be seen as a generalisation of the Bonferroni strategy. However, it is more intuitive: the constant ϵ corresponds to the

maximum expected number of false detections that one is ready to accept. It allows us to think in terms of number of false alarms (i.e., an absolute decision criterion that already takes into account the number of tests) rather than in terms of probability of false alarm (i.e., a statistical criterion). For example, consider NFA set with $\epsilon = 10$. If 1000 events are detected, then about 10 are false detections while the 990 remaining events are meaningful. This means that, if we were to increase the number of tested events, the number of errors would not increase and would remain limited to $\epsilon = 10$. This is not the case if we would have constrained the probability of false alarms instead.

The choice of ϵ depends on the task being handled. In the literature, the *a contrario* approach has been applied on several tasks such as point alignment or line segment detection. In these cases, the scene contains multiple objects, which means that many detections can be made in one image. Therefore, accepting a few false detections is not critical. As a simple convention, Desolneux et al. [30] suggest that using $\epsilon = 1$ is often sufficient. There are some contexts (e.g., medical or defense-related applications) where no false alarms can be accepted. For these cases, ϵ needs to be set extremely low, often less than 10^{-100}. In order to increase the readability of those values and to avoid rounding to zero problems, some authors (e.g., [75]) have rather considered the *significance* $S(\{x_k\}_{k\in G_l})$, defined using a logarithmic scaling:

$$S(\{x_k\}_{k\in G_l}) = -\ln\left(\mathrm{NFA}(\{x_k\}_{k\in G_l})\right). \tag{3.4}$$

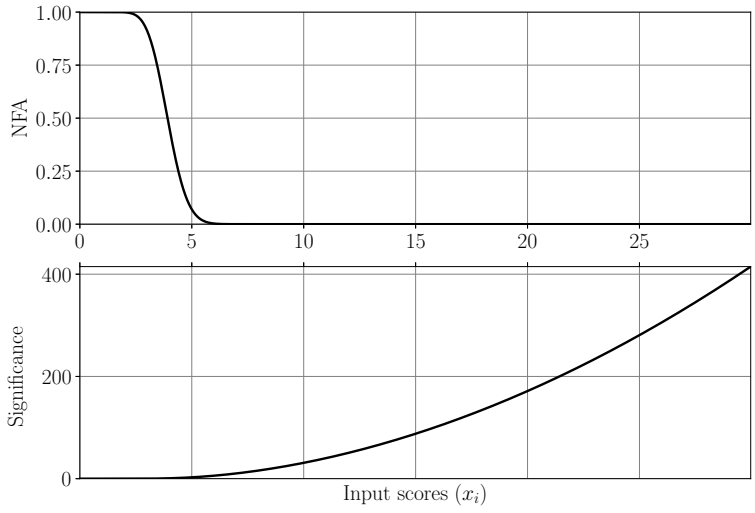


Figure 3.6: NFA and significance values for a centered and unit variance Gaussian variable. For simplicity, η_{test} is taken equal to 1.

The NFA values and their corresponding significance for a naive model following

a Gaussian distribution (defined in Section 3.4.1) are shown on Figure 3.6. The significance values range from $-\ln(\eta_{test})$ to $+\infty$, with large values corresponding to significantly unexpected objects.

Fusion of NFA maps In practice, in order to better detect objects of interest, we may be interested in applying the *a contrario* test to several versions of the same image (for example, with different preprocessing filters). In this way, we obtain several detection maps that need to be merged together in order to obtain a single map for each tested image. In this manuscript, we will consider the union of detections, whose definition is justified by the following property.

Property 3.3.2 (Weighted minimum of NFA)**.** Let X be a set of random variables following distribution H_0 and $G \in P(A)$.

If $\mathrm{NFA}_1, \mathrm{NFA}_2, ..., \mathrm{NFA}_N$ are NFA functions from $X \times P(A)$ to $\mathbb{R}^+$, for $N \in \mathbb{N}^*$, and $a_1, ..., a_N$ are positive real numbers satisfying $\sum_i a_j^{-1} \leq 1$, then $\min(a_1\mathrm{NFA}_1, ..., a_N\mathrm{NFA}_N)$ is also a NFA on $(X, P(A))$.

Proof. Let $a_1, ..., a_N$ be positive real numbers such that $\sum_j a_j^{-1} \leq 1$, and $F = \min(a_1\mathrm{NFA}_1, ..., a_N\mathrm{NFA}_N)$. For $x \in X$ and $\epsilon > 0$, we have:

$$\{G \in P(A), F(\{x_k\}_{k\in G_l}) \leq \epsilon\} = \bigcup_{j=1}^{N} \{G \in P(A), a_j\mathrm{NFA}_j(\{x_k\}_{k\in G_l}) \leq \epsilon\}.$$

By applying the cardinality function $\#\{.\}$, we have the following inequality:

$$\#\{\{G \in P(A), F(\{x_k\}_{k\in G_l}) \leq \epsilon\}\} \leq \sum_{j=1}^{N} \#\{\{G \in P(A), a_j\mathrm{NFA}_j(\{x_k\}_{k\in G_l}) \leq \epsilon\}\}.$$

The previous inequality can be rewritten as follows:

$$\#\{\{G \in P(A), F(\{x_k\}_{k\in G_l}) \leq \epsilon\}\} \leq \sum_{j=1}^{N} \#\{\{G \in P(A), \mathrm{NFA}_j(\{x_k\}_{k\in G_l}) \leq \frac{\epsilon}{a_j}\}\}.$$

Then, applying the expectation operator $\mathbb{E}[.]$ to previous equation, and using NFA property (3.2) on NFA_j functions,

$$\mathbb{E}[\#\{\{G \in P(A), F(\{x_k\}_{k\in G_l}) \leq \epsilon\}\}] \leq \sum_{l=1}^{N} \frac{\epsilon}{a_j} \leq \epsilon.$$

□

Finally, if $\mathrm{NFA}_1, \mathrm{NFA}_2, ..., \mathrm{NFA}_N$ provide N detection maps, we can obtain all the detected structures from $F = N \times \min_{j\in[\![1,N]\!]} \mathrm{NFA}_j$.

Multi-scale fusion of NFA maps - What if $\text{NFA}_1, \text{NFA}_2, ..., \text{NFA}_N$ have different spatial resolutions? This can be the case when considering the fusion of NFA maps that are computed at different spatial scales. In order to compute the minimum among all the NFA maps, it is necessary to resize the NFA maps so that they have the same spatial resolution. In the manuscript, we resize all the NFA maps to the highest spatial resolution. However, in this case, low resolution NFA maps may not have the same weight as high resolution NFA maps. Let $\eta_{test,1}, \eta_{test,2}, ..., \eta_{test,N}$ be the number of tests associated to $\text{NFA}_1, \text{NFA}_2, ..., \text{NFA}_N$, respectively. We assume that NFA_1 has the highest spatial resolution, NFA_N the lowest, and we introduce $r \leq 1$ the spatial downsampling ratio between NFA_i and NFA_{i+1}. In the case where $\eta_{test,i}$ is the number of pixels composing NFA_i, we have $\eta_{test,i+1} = r^2 \times \eta_{test,i} \leq \eta_{test,i}$. This may introduce an undesired bias towards low resolution maps since they may have a higher NFA and thus lead to less detections. To avoid a tricky discussion about how to weight the different spatial scales, as a basic approach, we set $\eta^*_{test} = \eta_{test,1} \sum_{k=0}^{N-1} r^{2k}$. We can easily verify that η^*_{test} satisfies the Property 3.3.1. Indeed, with the notations of Property 3.3.1, we have $\eta_l = \eta^*_{test}$, which boils down to verifying that, for all i, $\eta^*_{test} \geq \eta_{test,i}$. Since $\eta^*_{test} = \eta_{test,1} \sum_{k=0,k\neq i}^{N-1} r^{2k} + \eta_{test,1} r^{2i}$, and $\eta_{test,1} r^{2i} = \eta_{test,i}$, we effectively have $\eta^*_{test} \geq \eta_{test,i}$. Then, in the next chapters, we will propose a strategy in order to automatically tune a weighting coefficient for each scale in practice.

To conclude this section, let us summarise the main stages of the *a contrario* detection process:

- First, we set the ϵ value, the random variable(s) associated to the tested objects (attributes) and a measure function μ.
- Second, we choose a naive model that allows us to define our NFA test. This model often depends on the nature of the data (e.g., whether it is a binary image or a greyscale image). In the literature, the normal or binomial distributions are commonly used. Two practical examples will be given in Section 3.4.
- Third, we define the number of tests, which is the total number of objects tested.
- Finally, we test each object (after applying the measure function μ) using the NFA formula defined in Eq. (3.3). If we consider several NFA functions, they are merged using property 3.3.2. If the merged NFA value of the tested object is lower than ϵ, the background hypothesis is rejected and the tested object is labelled as detection.

In the next section, we give two concrete examples of *a contrario* detectors used as a post-processing step for infrared small target detection.

3.4 *A contrario* post-processing for small target detection

In this paragraph, we take the feature maps obtained in Section 3.1 and apply an *a contrario* thresholding to detect the targets. Our goal is to show the benefits of such thresholding over conventional thresholding. The following approaches were inspired by [31]. Their work consists of reducing any anomaly detection problem to a problem of detecting structures in noise, and then applying an *a contrario* criterion for detecting objects. The first stage of their approach computes the residual image by subtracting the background (self-similar elements) from the original image. The residual image is very similar to the feature maps extracted by the neural network, so, in our case, we directly apply the NFA test without having to extract a residual image. We propose to test two different *a contrario* criteria, which differ in the choice of the naive model. In the following, we present two *a contrario* formulations, one based on the normal distribution, the other on the uniform distribution.

3.4.1 Normal distribution as a naive hypothesis

The first naive model is the most straightforward for greyscale images (e.g., the feature maps we consider). Indeed, pixels that have high grey levels are considered likely to belong to a target. We thus make the naive assumption H_0 that the background noise follows gaussian distribution. The model does not need to be exact, the aim being to show significantly anomalous target values under the assumed model. Following Eq. (3.3), we define the NFA for any observation x as:

$$\mathrm{NFA}_{\mathcal{N}}(x) = \eta_{test} \times \frac{1}{s\sqrt{2\pi}} \int_x^{+\infty} \exp\left(\frac{(t-m)^2}{2s^2}\right) dt, \tag{3.5}$$

where m represents the mean and s the variance of the observed process.

In this paragraph, the *a contrario* testing is applied as a post-processing, after having extracted meaningful features from the image, i.e. features where the statistical testing will be rejected only for targets. In the literature, it is common to use conventional image processing methods such as filtering, as in [31]. In our case, the filtering step is replaced by a deep neural network. However, since a target is usually a few pixels wide, it may be interesting to gather neighbourhood information before statistical test. We thus apply $N_{conv} = 2$ additional convolutions with disks of different radii (1 and 2) as suggested in the article [31]. Note that by property, the background is still a white noise after convolution. In addition, the object detection problem we tackle is naturally multi-scale, since we wish to detect targets of different sizes. To do this, in addition to the different scales induced by

convolution with disks of different radii, we sub-sample the image by convolving with Gaussian kernels. We finally obtain $N_{channels} = 4$ scales, which gives us the following number of tests η_{test}:

$$\eta_{test} = N_{channels} \times N_{conv} \times \sum_{i=1}^{N_{scales}} N_{pixels}, \tag{3.6}$$

with N_{pixels} being the number of pixels composing the tested image, which has a resolution of 265×256 pixels. This leads to:

$$\eta_{test} = 4 \times 2 \times 256^2 \times (1 + \frac{1}{4} + \frac{1}{16} + \frac{1}{64}) \tag{3.7}$$

By applying this criterion to each scale, we end up with several NFA maps of different sizes. To carry out the detection, we oversample all the maps to obtain maps of size 256×256, then we take the minimum NFA value obtained on all the maps. The final score map is still a NFA according to property 3.3.2, which allows us to directly apply the threshold ϵ in order to perform the detection.

3.4.2 Uniform distribution as a naive hypothesis

One of the problems with the naive model presented in Section 3.4.1 is that spatial information (e.g., point density) is only modelled via the convolutions preceding the *a contrario* test. This information is not explicitly taken into account in the computation of NFA, which is performed at pixel level. However, point density is one of the essential perceptual criteria for discriminating targets from noisy background. Therefore, we propose to introduce an NFA criterion that estimates local point densities. A widely used naive model for this purpose considering binary maps is the uniform spatial distribution of the points, or "true" pixels, in the image lattice, leading to binomial distribution for the number of "true" pixels falling within any given parametric shape [30, 75]. In our case, as we deal with grey-level feature maps and not binary maps, we need to adapt the statistical testing, as well as the feature maps.

For the NFA criterion, we took inspiration from [76]. The main idea is to simultaneously consider grey level characteristics and spatial features (point density). We consider that a set of pixels likely to represent a target is all the more significant as it contains many points spatially close and of high value on the score map. We assume that the points are uniformly distributed within a bounded 3D space. The naive model is then the binomial distribution of parameter p representing the the pixels in a discrete and bounded 3D space, $\mathcal{E} \subset \mathbb{R}^3$, with axes representing the spatial coordinates and the transformed score values (third axis) (cf. next paragraph). The probability of observing at least κ pixels in a pavement of volume ν is then the Binomial distribution of parameter p and the NFA is written:

Algorithm 1 Detection of targets of maximum size M pixels on the score map I_s; input: I_s, M, minimal significance S_{min}; output: list of targets $\mathbf{C}$.

for for each pixel j in I_s **do**
    $I_s(j) = f(I_s(j))$
end for
$\mathcal{P} \leftarrow$ 3D point cloud derived from pixels $j / I_s(j) < +\infty$
$p \leftarrow \frac{|\mathcal{P}|}{\text{3D volume of } \mathcal{P}}$
Initialise an array Tab of dimension M to $+\infty$
Initialise an array Idx of dimension M to -1
for each 3D cuboid $\mathcal{C}$, of 2D projection (x,y) with an area smaller than M **do**
    $\kappa \leftarrow$ number of pixels in $\mathcal{C}$; $\nu \leftarrow$ volume of $\mathcal{C}$
    if $Tab[\kappa] > \nu$ **then**
        $Tab[\kappa] \leftarrow \nu$; $Idx[\kappa] \leftarrow$ index of $\mathcal{C}$
    end if
end for
for $\kappa \in [\![1, T]\!]$ **do**
    $\nu \leftarrow Tab[\kappa]$; $p_{\mathcal{C}} = \frac{\kappa}{\nu}$
    if $\nu < +\infty$ et $p_{\mathcal{C}} > p$ **then**
        compute $S(\mathcal{C})$ from Eq. (3.9)
    end if
end for
$\mathcal{I} \leftarrow$ list of index of cuboids sorted by $S(\mathcal{C})$
$S_{max} \leftarrow$ significance of the first element in $\mathcal{I}$
$\mathbf{C} \leftarrow \emptyset$
for each index i in $\mathcal{I}$ **do**
    $\mathcal{C}_i \leftarrow i^{th}$ cuboid according to $\mathcal{I}$
    if $S(\mathcal{C}_i) > S_{min}$ et $S(\mathcal{C}_i) > 0.8 \times S_{max}$ **then**
        $\mathbf{C} \leftarrow \mathbf{C} \cup \{\mathcal{C}_i\}$
    end if
end for

$$\mathrm{NFA}_{\mathcal{U}}(\kappa,\nu,p)=\eta_{test}\sum_{i=\kappa}^{\nu}\binom{\nu}{i}p^{i}(1-p)^{\nu-i}, \tag{3.8}$$

where η_{test} is the number of tests, here taken to be equal to the number of 3D blocks of the same size as the one considered in $\mathcal{E}$. In terms of practical implementation, three elements need to be specified.

Firstly, the score values are transformed so that low values are spread over a high dynamic range and high scores are concentrated over a low dynamic range. For this purpose, we used the inverse function: $\forall x>\tau, f(x)=\frac{1}{x-\tau}; \forall x\leq\tau, f(x)=+\infty$. In practice, the τ parameter makes it possible not to consider scores that are too low, thus reducing the algorithmic complexity.

Secondly, rather than computing the NFA of Eq. 3.8 which is numerically expensive, we use the significance defined by $S_{\mathcal{U}}(\kappa,\nu,p)=-\ln(\mathrm{NFA}_{\mathcal{U}}(\kappa,\nu,p))$ and the Hoeffding approximation: if $\frac{\kappa}{\nu}>p$,

$$S_{\mathcal{U}}(\kappa,\nu,p)\approx\nu\left[\frac{\kappa}{\nu}\ln\left(\frac{\frac{\kappa}{\nu}}{p}\right)+\left(1-\frac{\kappa}{\nu}\right)\ln\left(\frac{1-\frac{\kappa}{\nu}}{1-p}\right)\right]-\ln\eta_{test}. \tag{3.9}$$

Thirdly, to compute the number of points in each $\mathcal{E}$ block, we use the integral histogram as in [75]. Finally, Algorithm 1 summarises the main steps for computing the NFA with a trick that consists in computing the significance only for the most significant blocks *a priori* (those of minimal volume with a given number of points included).

3.4.3 Results obtained with a U-Net

Experimental set-up We consider the feature maps extracted by a U-Net, which is a usual segmentation network. The detector was trained on a part of MFIRST dataset, in the two different settings introduced in Section 3.1: 1) frugal setting, with only 100 training samples; 2) data-sufficient setting, with 2500 training images. We trained the U-Net in a frugal setting because its performance is particularly weak in this context. Indeed, the extracted feature maps are particularly noisy, and it can be interesting to see the benefits of the *a contrario* thresholding in such challenging conditions. More specifically, we compare three thresholding methods: 1) thresholding with a conventional fixed threshold of 0.5 (denoted "S" in the results), 2) *a contrario* thresholding, with the version based on the normal distribution ("$\mathrm{NFA}_{\mathcal{N}}$"), 3) thresholding with the second NFA test (uniform distribution, "$\mathrm{NFA}_{\mathcal{U}}$"). For the NFA tests, ϵ is chosen on a validation set so that it maximises the F1 score. We evaluate the detector and the different thresholding approaches on the MFIRST test set (100 images) in terms of precision, recall and F1 score calculated at object level.

Nb. training images	Threshold	Precision (%)	Recall (%)	F1 (%)
100	S	17.8	64.0	27.9
	$\text{NFA}_{\mathcal{N}}$	56.5	53.2	54.8
	$\text{NFA}_{\mathcal{U}}$	64.1	48.6	**55.3**
2500	S	65.9	80.6	72.5
	$\text{NFA}_{\mathcal{N}}$	81.2	77.7	79.4
	$\text{NFA}_{\mathcal{U}}$	95.4	73.6	**83.1**

Table 3.1: Performance of the U-Net trained on 100 or 2500 images. Predictions are given either by fixed thresholding (S), or after NFA testing ("$\text{NFA}_{\mathcal{N}}$" for the version based on Gaussian distribution, and "$\text{NFA}_{\mathcal{U}}$" for the one based on uniform distribution). Metrics are computed at object level. The *a contrario* test significantly improves the performance over fixed-value thresholding.

Results Table 3.1 shows the metrics obtained when evaluating the trained U-Net on a sufficient number of images (2500 training samples) and also in a frugal context (100 images). There is a noticeable difference in performance when comparing the U-Net with fixed thresholding to the one where we applied an NFA criterion. In particular, there has been a significant reduction in the number of false alarms, leading to an increase in precision values. However, this improvement comes at the expense of some good detections (reduced recall), but overall the NFA filtering remains beneficial, as shown by the F1 score. Its benefits are even greater in a frugal context: the F1 score is twice the baseline value. Now, comparing the two versions of NFA, the difference in performance between the two remains minimal compared to the results obtained with the fixed threshold, although the $\text{NFA}_{\mathcal{U}}$ version (naive model based on the binomial distribution) still seems to perform better. Although this version is more complex to compute, it seems that explicitly taking spatial information into account in the NFA formulation is beneficial for target detection.

Figure 3.7 shows some results that illustrate the benefits of an *a contrario* approach. The first column shows the original image, the second one the score map extracted by the U-Net, the third one the prediction obtained by applying a fixed threshold and the last one by applying an NFA criterion ($\text{NFA}_{\mathcal{U}}$ version). The predictions obtained by U-Net with a fixed threshold show numerous false positives induced by the background noise present in the score map (rows 2 and 3), although they are less perceptually relevant than the targets to be detected. These false positives are absent from the detections obtained after NFA test. We can also observe in the first row that some targets that are less visible in the score map are well detected with the *a contrario* approach, in contrast to the fixed threshold approach.

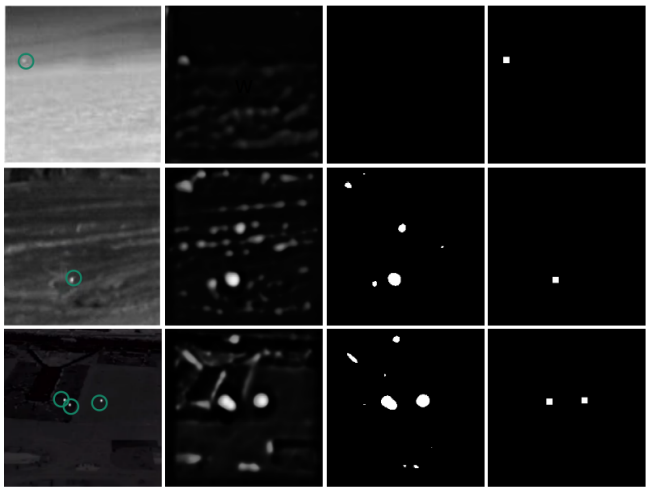


Figure 3.7: Examples of detector predictions on 3 images (rows). From left to right: original image with ground-truth circled, score map at network output, prediction after thresholding with a fixed threshold, prediction after applying $\text{NFA}_{\mathcal{U}}$ filtering.

Method		Precision	Recall	F1
U-Net	S	65.9	80.6	72.5
	$\text{NFA}_{\mathcal{U}}$	95.4	73.6	**83.1**
TransUnet	S	89.5	86.7	**88.0**
	$\text{NFA}_{\mathcal{U}}$	93.4	78.3	85.1

Table 3.2: Performance of the U-Net and TransUnet trained on 2500 images. Predictions are given either by fixed thresholding (S), or after $\text{NFA}_{\mathcal{U}}$ testing. Metrics are computed at object level. For TransUnet with $\text{NFA}_{\mathcal{U}}$ thresholding, the loss in good detections is not balanced by the increase in precision. Thus, applying an NFA test as a post-processing step has some limitations.

3.4.4 Discussion and conclusion

We have seen that for a detector that outputs very noisy score maps, the NFA thresholding is beneficial, especially with the version whose naive model is based on the binomial distribution. However, is the NFA thresholding still interesting when considering more accurate score maps? For this purpose, we consider a more efficient neural network, namely the TransUnet [42]. Compared to the UNet, this one has a ViT block in the extractor, which allows for a better feature extraction. We consider a TransUnet trained on 2500 images and compare fixed thresholding (S) and $\text{NFA}_{\mathcal{U}}$ thresholding. Based on Table 3.2, we can notice that TransUnet performs significantly better than UNet when a fixed threshold is applied to both networks. Applying *a contrario* test to a UNet allows us to get close

to the performance obtained with TransUnet. However, applying the NFA test to the TransUnet itself degrades performance: although the precision is slightly improved (i.e., the number of false alarms is reduced), the gain is not enough to compensate for the loss in good detection, leading to a decrease in the F1 score. The contribution of the NFA as a post-processing is therefore limited. Although the NFA can improve the detector's performance when the optimal parameters are chosen (threshold ϵ), this improvement will always be limited by the score map extracted by the network. This is especially true for networks trained on sufficient data, where the targets are not drowned in noise and false detections are due to a weak feature extraction process. Ultimately, the H_0 background model is not suitable for these cases. We need to be able to control the score map obtained by the detector so that the background follows a Gaussian or binomial distribution, where the target stands out from the noise, and where the NFA criterion would work better. To this end, in the next chapter, we propose to integrate a trainable NFA layer in the network in order to replace the fixed-value thresholding.

Chapter 4

Integrating *a contrario* decision criterion into segmentation networks

In this chapter, we propose to integrate an *a contrario* decision criterion into the training loop of a segmentation network. Although segmentation networks are not optimal for object detection compared to object detectors (indeed, a post-processing step is necessary to perform the detection of objects, unlike with YOLO or Faster-RCNN), we decide to consider segmentation networks because they are SOTA for infrared small target detection. This can be explained by the fact that they reconstruct the features at a higher spatial resolution, which benefits the detection of small objects.

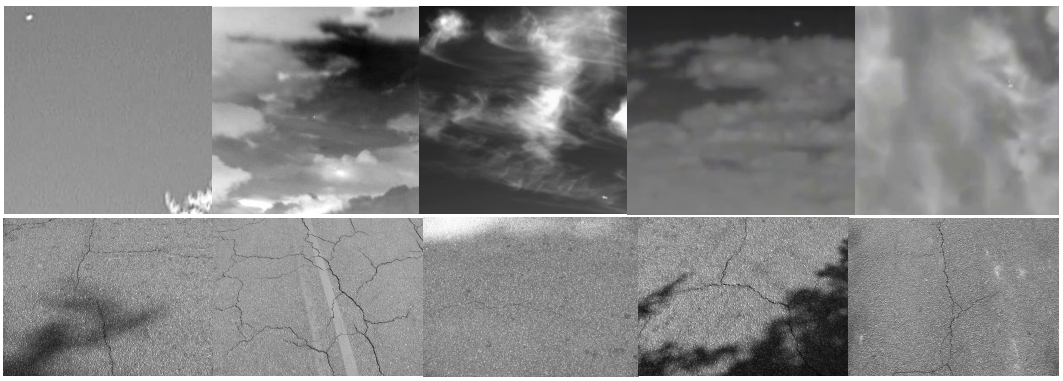


Figure 4.1: Example of tiny objects. The first line shows small targets on a sky background. Note the challenging conditions: very small targets, low contrast, cloud-induced textures. The second line shows road cracks, which have different thicknesses, and are sometimes blended with the textured roads or shadows.

However, these NN still struggle with small object detection. This is firstly due to the nature of the objects: their surface area is made of only few pixels, and they

do not present a specific structure. Secondly, tiny objects are often partially hidden in complex and highly textured backgrounds, leading to many false alarms. Some examples are shown on Figure 4.1. Thirdly, dealing with small object detection results in learning from highly class-imbalanced datasets. Indeed, as there are very few target pixels compared to the background pixels, errors made on the targets have less impact on the training than the ones made on the background. Therefore, tiny object features cannot be learned easily. As detailed in Chapter 2 Section 2.4, some methods attempt to artificially balance the dataset, either by using data augmentation or by introducing weights into the training loss depending on the class of the object (e.g., focal loss [27]).

We propose to go further and to take advantage of the *unexpectedness* of small objects with respect to the background, as one could do in an anomaly detection approach with, for example, one-class classifiers [29], that discriminate small objects as *unexpected* patterns with respect to the background. These methods are typically trained on anomaly-free samples to model the background distribution, with objects deviating from this distribution during inference classified as anomalies. Training is conducted in a "weakly" supervised manner, as it relies exclusively on "normal" samples. However, in our case, we opt for conventional supervised training since our datasets include target samples, and excluding them from the training process would be a missed opportunity. To leverage this, we propose incorporating an anomaly *a priori* specific to small targets into the supervised training process by utilizing *a contrario* reasoning. For this purpose, we introduce the *a contrario* criterion presented in Section 3.4.1. In the latter, we have seen that using this criterion as a post-processing appears suboptimal since the feature map statistical distribution may not match the naive assumption made on the background when applying *a contrario* decision. We therefore propose to guide the NN training by including the *a contrario* criterion in the training loop through our NFA module. The latter guides the network to extract features in a way that the object features will be likely to contradict the naive hypothesis made on the background. We specifically choose the $\text{NFA}_{\mathcal{N}}$ formulation proposed in Section 3.4.1 since it is simple to compute and it provides a *significance* score at a pixel-level, which is relevant for semantic segmentation. We propose to integrate it as a block with a specific activation function, which can replace the segmentation head of any one-class segmentation NN. Its integration within a U-shaped NN is presented in Figure 4.2. In the following, we first provide theoretical details about our $\text{NFA}_{\mathcal{N}}$ formulation and the associated components so that it can be integrated into the network and trained in an end-to-end manner. Then, we evaluate the benefits of our method for small target detection and show that it leads to competitive performance for IRSTD while being more interpretable. Finally, we extend the use of our method on two other challenging tasks, namely ship detection from remote

sensing data and road crack detection.

4.1 Methodology

4.1.1 Multi-channel formulation

Let us first provide more details about the $\text{NFA}_{\mathcal{N}}$ formulation. As we deal with multi-channel feature maps, we need to adapt the $\text{NFA}_{\mathcal{N}}$ formulation introduced in Section 3.4.1 since it is designed for single-channel images. In [31], the authors adapted the single channel formulation to multi-channel input by considering each channel independently. The obtained NFA maps are then merged together by taking the union of detections. In the following, we rather reformulate the previous approach in terms of a multivariate normal distribution, as suggested by [77]. By considering a centred input X_i with K channels, we can rewrite Eq. (3.3) using the Gamma and upper incomplete Gamma functions (denoted $\Gamma(.)$ and $\Gamma(.,.)$ respectively):

$$\text{NFA}_{\mathcal{N}}(x_i, \eta_{test}, K, \Sigma) = \frac{\eta_{test}}{\Gamma(K/2)} \Gamma\left(\frac{K}{2}, \frac{1}{2}||\Sigma^{-1/2} x_i||_2^2\right), \tag{4.1}$$

where Σ represents the covariance matrix of the centred variable X_i. Note that Eq. (4.1) is a multi-channel generalisation of Eq. (3.5), and therefore we keep the same notation. In the remainder of the manuscript, $\text{NFA}_{\mathcal{N}}$ will refer to Eq. (4.1).

Three assumptions about the feature noise can then be considered: 1) Elliptical distribution with dependent channels: in this case, Σ is a dense positive-definite matrix; 2) Elliptical distribution with independent channels, which leads to $\Sigma = \lambda\Delta$ where Δ is a diagonal positive matrix with $|\Delta| = 1$ and $\lambda \in \mathbb{R}^{+*}$; 3) Spherical distribution, leading to $\Sigma = \lambda I_d$ where I_d is the identity matrix. In this particular case, no direction or channel is privileged in the decision process. The impact of these different hypotheses on the training is assessed in Section 4.2.3.

Then, the *significance* associated to Eq. (4.1) is:

$$S_{\mathcal{N}}(x_i, \eta_{test}, K, \Sigma) = -\ln\left(\frac{\eta_{test}}{\Gamma(K/2)} \Gamma\left(\frac{K}{2}, \frac{1}{2}||\Sigma^{-1/2} x_i||_2^2\right)\right). \tag{4.2}$$

NFA values and their corresponding *significance* are represented on Figure 3.6. Note that due to rounding problems to 0, we use the approximation of the $\Gamma(a, x)$ function for $x \to +\infty$ (in practice, for $x > 40$) given in [78]:

$$\Gamma(a, x) \approx x^{a-1} e^{-x} \left(1 + \frac{a-1}{x} + \frac{(a-1)(a-2)}{x^2}\right). \tag{4.3}$$

4.1.2 Deep-learning based NFA block

After adapting our NFA formulation to multi-channel feature maps, we now focus on the practical integration of this criterion within a deep learning framework. There are several challenges, including implementing a derivable version of the NFA criterion in order to perform the backpropagation to train the network, and also designing a specific activation function allowing for the use of any conventional cost function. Indeed, the *significance* scores provided by our NFA statistical test differ from conventional NN outputs. We propose a basic version of our NFA block, as well as a version that includes spatial attention mechanisms, which are useful for detecting small objects with various shapes. We also propose a NFA fusion block that enables deep supervision as in [79] (i.e., guiding intermediate NFA layers during the training).

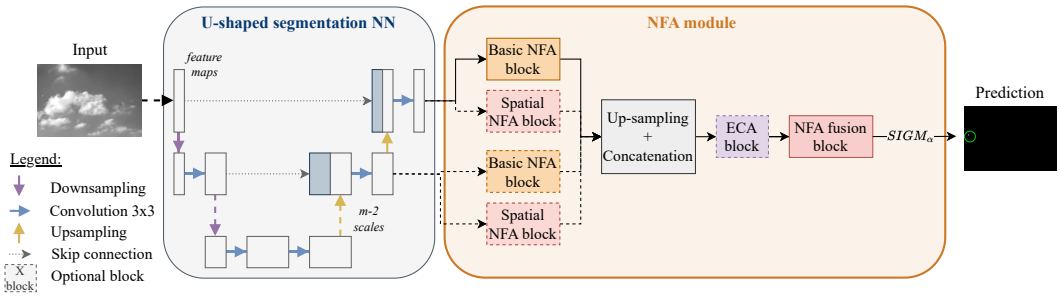


Figure 4.2: Diagram showing the integration of our NFA module into a U-shaped segmentation NN. Optional blocks are drawn in dotted lines. Details for ECA block can be found in the original paper [13].

Basic NFA block

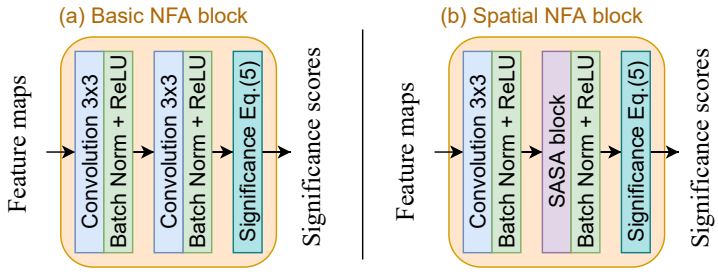


Figure 4.3: Diagram of (a) the basic NFA block and (b) the spatial NFA block. The details of the stand-alone self-attention (SASA) block can be found in [14].

We propose a basic NFA block that transforms multi-channel feature maps into a one-channel score map representing the *significance* defined by Eq. (4.2). This block is described by Figure 4.3a. Two convolution blocks (i.e., 2D convolution with kernel 3×3 followed by batch normalisation and ReLU activation) are applied on the input features in order to extract some relevant features for computing the NFA. The *significance* scores are then computed using Eq. (4.2), where η_{test} is equal to the total number of tested pixels for a given image (i.e., the size of the image). This equation is derivable, allowing for the backpropagation step in the NN.

Our NFA block can replace the segmentation head of any one-class segmentation NN. Its integration on a U-shaped NN is presented in Figure 4.2. Introducing the *a contrario* criterion into the supervised training loop will guide the network to extract features in a way that object features will be likely to contradict the naive hypothesis (here Gaussian distribution) made on the background.

Add attention mechanisms

The basic NFA block defined previously is designed to improve the detection of tiny objects that do not present a specific geometric structure (e.g., point-shaped objects). However, for objects that are small only in one dimensionality and large in other dimension(s) (e.g., cracks), spatial information is a discriminating feature. Indeed, in the case of crack detection, several pixels forming a continuous line are more likely to belong to a crack than a few isolated pixels. It is therefore necessary to extend the NN receptive field in order to better take into account the information from more distant pixels.

To improve performance on such objects, we design a second version of our NFA block that includes spatial attention mechanisms. Spatial attention is intended to indicate the regions of the image where most attention is needed. This is achieved by modelling long-range dependencies between the different regions of an input. Such attention can be useful for detecting objects that are tiny in one dimension and medium to large in the other (e.g., cracks). Several strategies have been proposed, including training a subnetwork to identify the important regions [80], or increasing the receptive field of CNNs. The methods based on the latest strategy use self-attention mechanisms, which were introduced in computer vision tasks by [81]. They lead to impressive results compared to the performance achieved so far using CNNs, especially when it comes to the use of ViT for various visual tasks [41]. However, this process is highly computationally expensive, and it also requires a lot of training data. In addition, it is interesting to note that the spatial dependencies are mainly local for small object detection. In this work, we rather consider the use of local self-attention layers, and more specifically the stand-alone self-attention layers proposed by [14].

Figure 4.3b shows this block, where the second convolution layer is replaced by a stand-alone self-attention (SASA) layer [14]. As shown in Figure 4.2, if we add a spatial NFA block, it is done in addition to a basic NFA block.

Multi-scale fusion of *significance* maps

Many popular segmentation networks rely on encoder-decoder models introduced in [82, 5]. The advantage of using U-shaped NN is that we can easily extract low-level semantic feature maps and use the large-scale spatial information they contain for detecting objects of different sizes. Although the highest-level feature maps are the most relevant for segmenting tiny objects, we will see that the feature maps from deeper scales are also useful. These contain rich spatial information and enable the NN to detect targets of different sizes (and therefore not be specific to a single target size). They also allow us to better discriminate targets from potential background false alarms. To do so, we integrate our basic NFA block at each intermediate scale of any U-shaped NN, as illustrated in Figure 4.2. Considering a NN with m scales, we perform the detection at each scale and thus obtain m *significance* score maps. Based on the discussion about multi-scale fusion of NFA maps in Section 3.3 of Chapter 3, we introduce $\eta_{test} = h \times w \times (1 + \frac{1}{2^2} + ... + \frac{1}{2^{2(m-1)}})$, where $h \times w$ is the number of pixels composing the image. In order to merge the detections performed at all scales, the low-level *significance* score maps are upsampled to match the NN input size $h \times w$ using bilinear interpolation. All *significance* maps $S_1, ..., S_m$ are then merged together through the NFA fusion block by taking the union of all detections. This leads to the final *significance* score map S_{final}, defined for each pixel i as follows:

$$S_{final}(i) = \max\{S_1(i), ..., S_m(i)\}. \tag{4.4}$$

However, with such a multi-scaling strategy, the detections from the lower and higher resolution scales have the same weight in the final *significance* score map, which may increase the false alarm rate for applications where coarse scales are less relevant. We thus propose to dynamically weight the impact of the different scales by learning weighting coefficients using a channel attention module.

Channel-based attention allows us to select the relevant channels in a set of feature maps. This concept was firstly presented in [83], where the authors introduce a squeeze-and-excitation block made of two steps. The first one, called the squeeze step, consists in a reduction in dimensionality while keeping global spatial information. Then, an excitation module allows for learning channel-wise relationships, which gives rise to an attention vector that indicates the weights to apply to the different channels. Several variants have been proposed to overcome SE block shortcomings. For example, [13] propose the Efficient Channel Attention (ECA) block, where they reduce the complexity of the fully-connected layers used

in the excitation step by replacing them with a 1D convolution. In the following, we focus on this solution. The integration of an ECA block [13] before merging the *significance* maps is illustrated on Figure 4.2.

NFA-friendly activation function

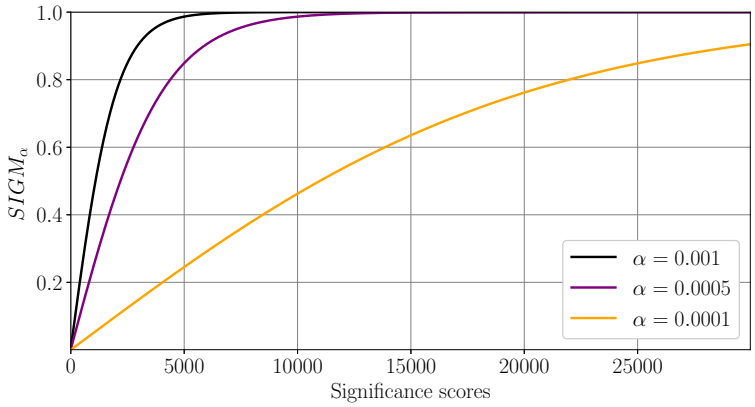


Figure 4.4: Variations of SIGM_α function defined in Eq. (4.5), with different values of α. For simplicity, we choose $\eta_{test} = 1$.

The NFA block output is a *significance* score map whose distribution of scores is not only asymmetric between positive and negative values, but also has a much wider dynamic than conventional NN output. Indeed, as explained in Section 3.3 (and illustrated by Figure 3.6), the background values are expected to be pushed towards $-\ln(\eta_{test}) \leq 0$, while the object values are expected to be spread over the interval $(-\ln(\eta_{test}), +\infty)$. Consequently, the conventional symmetric activation functions, such as the sigmoid function, are not suitable. This has been confirmed by the experiments presented in Section 4.2.3. Therefore we rather design the following activation function:

$$\text{SIGM}_\alpha(x, \eta_{test}) = \frac{2}{1 + e^{-\alpha(x+\ln(\eta_{test}))}} - 1, \tag{4.5}$$

where $\alpha \in \mathbb{R}^{+*}$ is a parameter that allows us to control the slope of the sigmoid. We represent the variations of SIGM_α function on Figure 4.4 for different values of α. This activation function strongly penalises the background values while compressing progressively the object values, thus respecting the dynamics induced by the computation of the *significance*. Note that the higher the value of the parameter α, the more the dynamic of the *significance* scores will be non linearly compressed. The sensitivity of the NN training to this parameter is studied in Section 4.2.3. This activation function is applied after having combined all the

significance maps obtained from the NFA blocks computed at different scales, as shown on Figure 4.2. The final output scores, therefore, range between 0 and 1, which allows the user to apply any cost function that is suitable for one-class segmentation tasks.

Nevertheless, substituting the conventional segmentation head in a segmentation NN by the NFA module makes the threshold usually used to binarise the segmentation map (namely, 0.5) no longer suitable, as background values are constrained to 0 after applying the function SIGM_α and even low output values can be significant. Thus we have to derive the new detection threshold. One argument often given in favour of *a contrario* approaches is the interpretation of the NFA and thus the more or less direct choice of the threshold (nevertheless application dependent). In our case, the segmentation threshold t is linked to the ϵ threshold defined in Eq. (3.2) through Eq. (4.5):

$$t = \text{SIGM}_\alpha(-\ln(\epsilon), \eta_{test}), \tag{4.6}$$

where ϵ represents the average number of false alarms on background images, at pixel level, that we can tolerate for our application. In the literature, ϵ is chosen in the interval $[10^{-200}, 1]$, leading to a thin threshold interval for the value t, namely $[10^{-3}, 0.12]$. We will discuss a more refined choice of this theoretical threshold for each considered application, depending on our tolerance for false alarms.

In the following section, we consider different backbones and evaluate the benefits of our NFA module on three applications, namely small target detection, road crack detection and ship detection. Note, however, that the former has been studied more extensively, and the latter two have been considered mainly to test the possible generalisation of the NFA module to other applications (and backbones).

4.2 Application to small target detection

We first evaluate the contribution of our NFA module in the case of infrared small target detection. This application constitutes an ideal framework for the detection of tiny objects: the targets have a surface area of only a few pixels, are not very contrasted compared to the background and do not present a specific structure. Most proposed methods to tackle this problem use semantic segmentation NN [15] rather than off-the-shelf detection NN [4]. SOTA NN for small target detection rely on U-shaped architectures and include spatial attention mechanisms [8, 7, 84].

4.2.1 Assessed methods

We propose to integrate our NFA module into one of the U-shaped SOTA backbones. We select the recent DNANet [8], which has shown impressive performance

on widely used small target detection datasets. DNANet is composed of two parts: a dense-nested U-shaped backbone (DNIM), which allows for the feature extraction step, and a feature pyramid fusion module (FPFM), which allows for a multi-scale fusion of intermediate outputs from the backbone. We substitute the FPFM block with our NFA module, which becomes DNIM + $\text{NFA}_{\mathcal{N}}$, and we evaluate its contribution with respect to the backbone DNIM (ResNet-18 version) and DNANet. We also extend our experiments to the use of a classical backbone, namely ResUNet [85], to show the generalisation of our NFA block to another backbone that is not specifically designed for small target detection. Results from other infrared small target detection baselines, including ACM [6], LSPM [26], AGPCNet [64] and MTU-Net [65], are provided in order to assess our methods. For the first three methods, we used the implementation given by kourenke/Review-Infrared-small-target-segmentation-networks. For MTU-Net, we used the original GitHub repository TianhaoWu16/Multi-level-TransUNet-for-Space-based-Infrared-Tiny-ship-Detection. These methods are trained from scratch using the parameters provided by the original papers, except for AGPCNet for which the Adam optimiser with a learning rate of 0.001 works best.

For our NFA module, we set the α parameter in Eq. (4.5) to 0.0005, as it has shown to lead to the best results in Section 4.2.3. To guide the selection of the binarisation threshold, let us remind that the considered application handles detections at object level (the first parameter of interest is the number of detected targets and then their localisation, speed etc.). Thus, the impact of one-pixel false alarm is completely different whether it is isolated or connected to a detection, since it will or not affect the number of detected targets. The strong constraint to absolutely avoid such errors implies a very low tolerance for false alarms at pixel level. In the literature, a low NFA in the case of the *a contrario* approach lies around $\epsilon \approx 10^{-200}$, leading to a binarisation threshold $t \approx 0.1$. This value has been confirmed on a validation set and we kept it unchanged for every experiment of this application. For the baselines, the detection threshold is set to 0.5 as suggested in the original paper. DNIM, DNANet, and DNIM + $\text{NFA}_{\mathcal{N}}$ are trained from scratch[1] on Nvidia RTX6000 GPU for 1000 epochs using the Soft-IoU loss function [58]. The latter is optimised by Adagrad optimiser with the Cosine Annealing scheduler, using the same parameters as in [8]. The learning rate is set to 0.05 for DNIM and DNANet as suggested in the original paper. For DNIM + $\text{NFA}_{\mathcal{N}}$, we found that decreasing the learning rate allows for better convergence; we thus set it to 0.03.

[1] We used the official implementation of DNANet https://github.com/YeRen123455/Infrared-Small-Target-Detection

4.2.2 Dataset and evaluation metrics

We conduct our experiments on two datasets. We first consider NUAA-SIRST dataset [6], which is one of the few infrared small target datasets publicly released and widely used in the literature. This dataset is described in Section 2.3.2. We also consider a recently published dataset for small target detection, namely IRSTD-1k [7]. As described in Section 2.3.3, this dataset is larger (1000 images) and contains more challenging scenes, with different kinds of small objects (e.g., aircrafts, animals). It also contains some very large objects, which fall outside the scope of our method (designed for tiny object detection, cf. Section 4.4). We therefore consider the "IRSTD-850" version of this dataset (where large targets are removed, as explained in Section 2.3.3). We discuss the behaviour of our method on larger objects in Section 4.4. Both datasets are split into training, validation and test sets using a ratio of $60:20:20$. We use the same pre-processing steps as those proposed in [8]. We also resize all images to the resolution 256×256.

For the evaluation, we mainly focus on the object-level metrics as suggested by [8]. From the predicted binary segmentation map, targets are individually labelled using a 8-connectivity connected component module. A detected object is counted as a true positive (TP) if it has an Intersection over Union (IoU) of at least 5% with the ground truth. This low-constrained condition is due to the fact that a small shift in the number of predicted pixels leads to a large deviation in the IoU, as illustrated in [66] (Fig.1). We then compute the Precision (Prec.), Recall (Rec.) and F1 score (F1) at object-scale. We also consider the area under the object-level Precision-Recall curve, namely the AP, which allows us to free from the detection threshold, and the number of False Alarm (FA) (still at the object-level) per image (FA/image).

In the tables, the presented results have been averaged over five distinct training sessions for DNIM, DNANet and DNIM + $\text{NFA}_{\mathcal{N}}$, and they are given in the form $m \pm s$, where m is the mean and s the standard deviation, calculated over the different training sessions.

4.2.3 Results

a) NFA module improves the precision In this paragraph, we compare the performance of our method against SOTA networks for infrared small target detection. We also provide the results obtained when using a conventional segmentation network as a backbone.

Results obtained with SOTA backbones - Table 4.1 shows the performance for the compared methods on NUAA-SIRST and IRSTD-850 datasets. First, we can see that DNIM+$\text{NFA}_{\mathcal{N}}$ outperforms most of the IRSTD baselines (including ACM, LSPM, AGPCNEt and MTU-Net) by a wide margin, and is on a par

Method	F1	AP	Prec.	Rec.	FA/image
NUAA-SIRST dataset					
ACM [6]	95.4	95.2	95.1	95.8	-
LSPM [26]	92.9	90.2	90.3	95.9	-
AGPCNet [64]	93.8	92.2	93.8	95.1	0.07
MTU-Net [65]	93.8	97.2	92.9	94.8	0.08
DNIM	$95.8^{\pm 1.3}$	$96.2^{\pm 1.3}$	94.6	97.1	$0.06^{\pm 0.03}$
DNIM + FPFM (DNANet)	$\underline{97.1}^{\pm 0.4}$	$\mathbf{98.4}^{\pm 0.9}$	$\underline{96.9}$	$\underline{97.3}$	$\underline{0.04}^{\pm 0.02}$
DNIM + NFA$_{\mathcal{N}}$	$\mathbf{97.6}^{\pm 0.3}$	$\mathbf{98.4}^{\pm 0.6}$	**97.9**	**97.4**	$\mathbf{0.02}^{\pm 0.00}$
IRSTD-850 dataset					
ACM	62.1	48.4	62.4	61.9	0.55
LSPM	54.9	51.5	64.9	47.6	0.38
AGPCNet	88.1	92.3	91.1	85.3	**0.12**
MTU-Net	86.8	89.0	88.8	84.9	0.16
DNIM	$89.0^{\pm 1.4}$	$89.9^{\pm 1.6}$	87.6	90.5	$0.20^{\pm 0.05}$
DNIM + FPFM (DNANet)	$\mathbf{91.4}^{\pm 1.4}$	$\underline{92.4}^{\pm 1.9}$	$\underline{91.8}$	**91.1**	$\underline{0.13}^{\pm 0.04}$
DNIM + NFA$_{\mathcal{N}}$	$\underline{91.3}^{\pm 0.7}$	$\mathbf{94.2}^{\pm 0.2}$	**92.1**	$\underline{90.6}$	$\mathbf{0.12}^{\pm 0.00}$

Table 4.1: Object-level F1 (%), AP (%), Prec. (%), Rec. (%), and FA/image achieved by the compared methods on NUAA-SIRST and IRSTD-850. For each dataset, best results are in bold and second best results are underlined.

with the SOTA method DNANet. The performance gap between LSPM, MTU-Net and the other IRSTD baselines can be explained by the fact that: i) most methods are designed to achieve good pixel-level segmentation, which does not necessarily translate into good object-level performance, and ii) the SIRST dataset contains very little data, and MTU-Net may be suboptimal in a frugal setting because it contains ViT blocks in its encoder (which are known to require a large training dataset).

Second, it can be noticed that DNIM+NFA$_{\mathcal{N}}$ leads to a significant improvement of the baseline DNIM in both AP and F1, on both datasets. For example, the F1 score is increased by 1.8% on NUAA-SIRST dataset and by 2.3% on IRSTD-850. More specifically, since the NFA layer controls the number of false alarms, the precision appears significantly improved, while keeping the number of correctly detected targets (recall criterion) at the same level. This improvement in precision is all the more impressive on the challenging IRSTD-850 dataset (+4.3% in AP). Note that the addition of the NFA module in DNIM greatly improves the stability of the training, as evidenced by the decrease in the standard deviation of

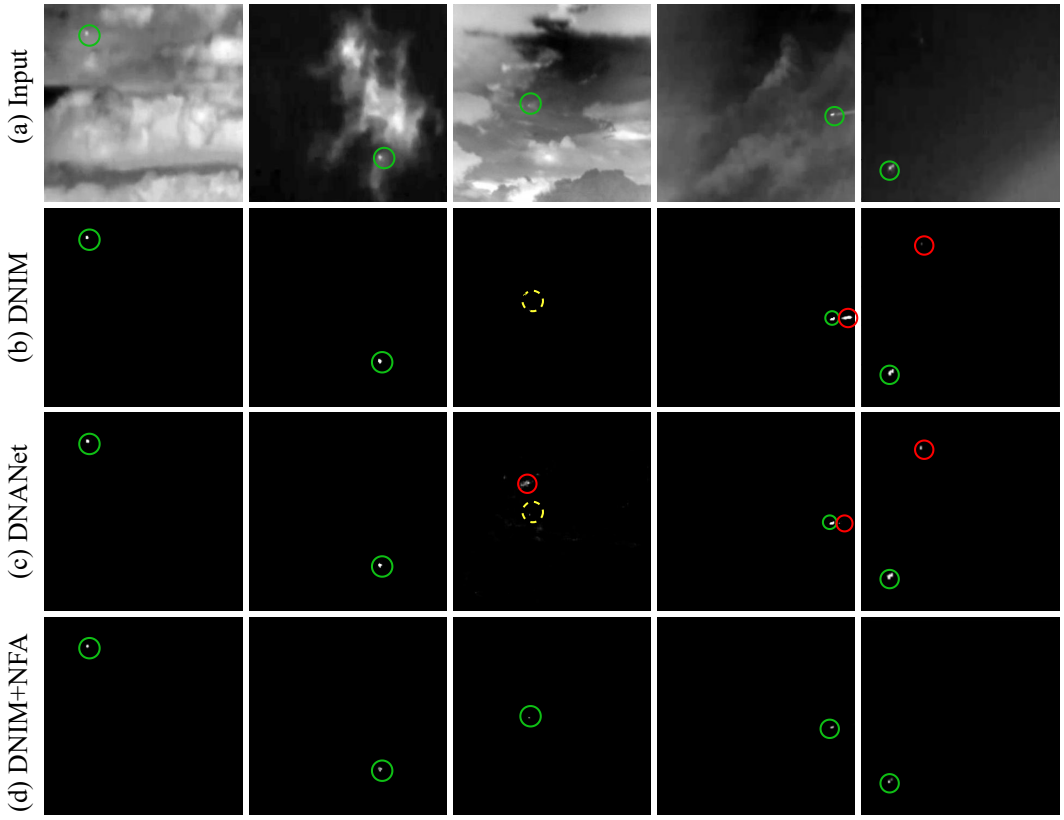


Figure 4.5: Qualitative results obtained with different detection methods (columns (b) to (d)) on NUAA-SIRST dataset. Good detections, false positives and missed detections are circled in green, red and dotted yellow lines respectively.

the results. DNIM+NFA$_{\mathcal{N}}$ is also very competitive with SOTA method DNANet. Indeed, on NUAA-SIRST dataset, the F1 score is better in average (+0.5%), and it can be noticed that the number of false alarms per image has been divided by 2, while having a better recall. The standard deviation is also reduced. On IRSTD-850 dataset, although the F1 scores are equivalent for both methods, the AP is significantly improved by DNIM+NFA$_{\mathcal{N}}$ (+1.8%). This confirms the benefit of our NFA module, especially on the control of the false alarm rate even in scenes with complex backgrounds. Furthermore, as far as computation costs are concerned, the NFA layer adds less than 0.1 million training parameters to the initial model, which is negligible with respect to the benefits deriving therefrom.

Figure 4.5 illustrates some predictions (output score maps before threshold) on challenging scenes, where the contribution of the NFA module can clearly be seen. For example, the target of the third column is particularly small and blurred in the background, which does not affect the performance of the NFA module, unlike

Method	*NUAA-SIRST*		*IRSTD-850*	
	F1	**AP**	**F1**	**AP**
ResUNet	$93.2^{\pm 0.9}$	$90.3^{\pm 2.4}$	$85.3^{\pm 1.2}$	$87.4^{\pm 1.2}$
ResUNet + $\text{NFA}_{\mathcal{N}}$	$\mathbf{95.4}^{\pm 1.3}$	$\mathbf{96.1}^{\pm 1.9}$	$\mathbf{87.7}^{\pm 2.7}$	$\mathbf{96.0}^{\pm 0.8}$

Table 4.2: Comparison of ResUNet and ResUNet + NFA on small target detection. Metrics are computed at object-level and averaged over three runs.

the other methods. Moreover, the baseline methods mistakenly detect the aircraft contrail (fourth column). The NFA module not only allows for better detection of small and tiny objects in particularly difficult scenes, but also provides robustness with respect to challenging environments.

Generalisation to conventional backbones - We extend the experiments conducted previously to another conventional NN, namely ResUNet. We evaluate the benefits of our NFA module for a NN that is not specifically designed for small target detection. Results are presented in Table 4.2, and it can be seen that the NFA module greatly improves global performance. Indeed, on NUAA-SIRST dataset, the F1 score is improved by 2%, and the AP by 6%, which is mainly explained by an increase in average precision as observed for DNIM+$\text{NFA}_{\mathcal{N}}$. This improvement in precision is even more striking when considering IRSTD-850 dataset (+8.6% in AP). This confirms that adding an NFA module on a segmentation network (being SOTA or not) improves the precision and thus the performance. Furthermore, the results obtained when adding the NFA module on a conventional segmentation NN are only few percents lower than what can be obtained by a SOTA backbone specifically designed for small target detection. For example, the difference in F1 score between ResUNet+$\text{NFA}_{\mathcal{N}}$ and DNIM is only of 0.4% on NUAA-SIRST. This shows that although the careful design of the feature extractor is essential to improve performance, the choice of decision criterion is also very important, especially in the case of small/tiny object detection.

b) Overconfidence, did you say? Most recent neural networks tend to be overconfident as outlined in [86]. The pixel-level histogram of output scores shown on Figure 4.6a illustrates this phenomenon for DNIM network, where all pixel values on the final score map are either very close to 0 or to 1. The impact of NFA layer can clearly be seen on the corresponding histogram in Figure 4.6b: TP are uniformly spread all over the confidence scores and the number of false positives (FP) decreases monotonically as the score level increases. Figures 4.7 illustrate the relationships between accuracy and output scores (interpreted as confidence values). When comparing Figure 4.7a and Figure 4.7b, we see that the achieved scores are more informative since the accuracy versus score function is

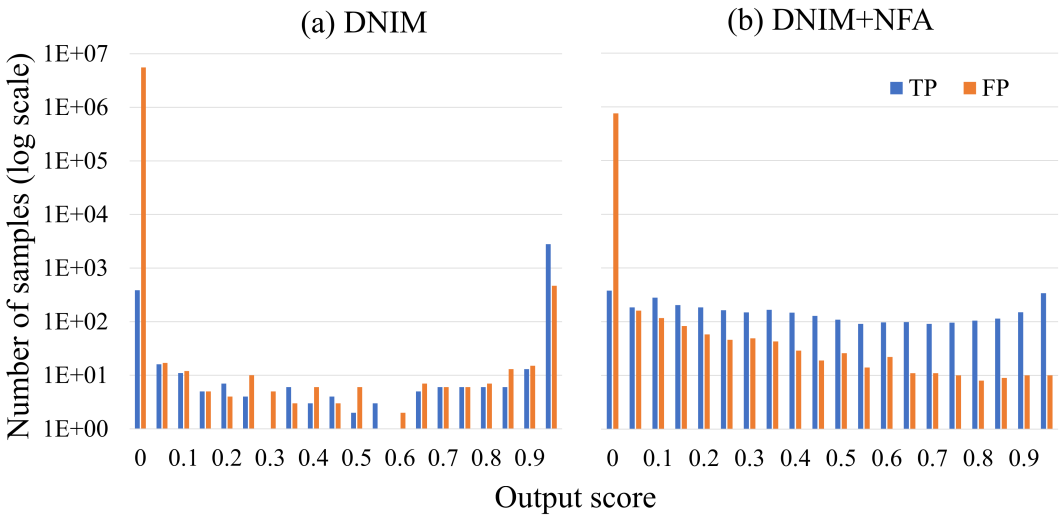


Figure 4.6: Output scores histograms for (a) DNIM and (b) DNIM+NFA$_{\mathcal{N}}$.

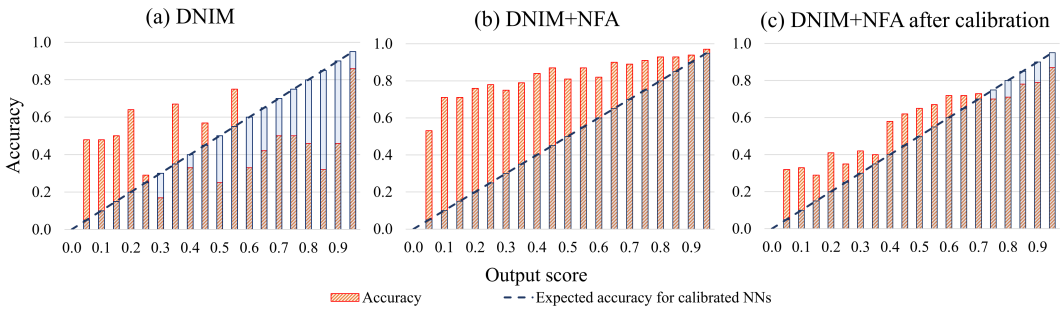


Figure 4.7: Variations in accuracy as a function of output scores for (a) DNIM, (b) DNIM+NFA$_{\mathcal{N}}$ with $\alpha = 0.0005$, and for (c) DNIM+NFA$_{\mathcal{N}}$ after calibration using $\alpha = 0.003$.

globally increasing. Besides, we notice that the NFA module prevents the network from being overconfident. To better calibrate the DNIM+NFA$_{\mathcal{N}}$ outputs, we can fit *a posteriori* the parameter α from Eq. (4.5). The value of α to get a calibrated network can be found by solving $\text{SIGM}_{\alpha}(10^{-200}, \eta_{test}) = 0.5$ (since for a calibrated output optimal segmentation threshold should be equal to 0.5), so that $\alpha = 0.003$. Figure 4.7c illustrates the results after calibration using $\alpha = 0.003$: adding the NFA module to a segmentation NN allows us to obtain a nearly calibrated network without the need of complex methods. The output scores are much more relevant than with the baseline, which is a step towards Artificial Intelligence (AI) interpretability.

Method	15-shots		25-shots	
	F1	**AP**	**F1**	**AP**
DNIM	$72.8^{\pm 23.7}$	$68.0^{\pm 31.3}$	$87.0^{\pm 2.5}$	$82.6^{\pm 2.7}$
DNIM + $\text{NFA}_{\mathcal{N}}$	$\mathbf{87.7}^{\pm 2.9}$	$\mathbf{86.3}^{\pm 3.9}$	$\mathbf{90.9}^{\pm 2.7}$	$\mathbf{93.1}^{\pm 2.0}$

Table 4.3: Results achieved in 15 and 25-shot settings on NUAA-SIRST. Best results are in bold.

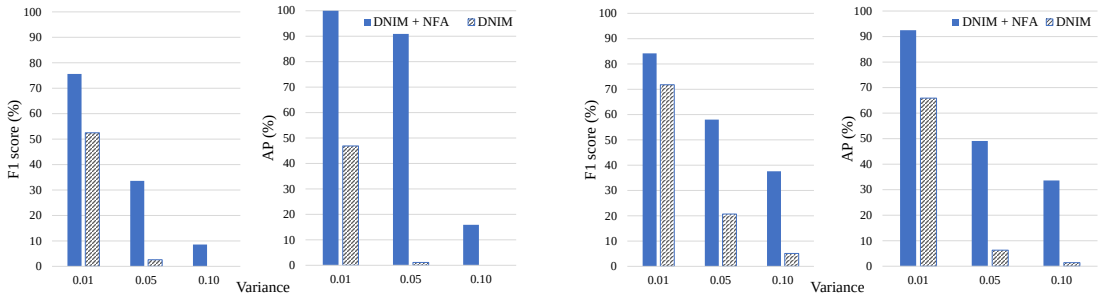
Method	F1	AP
DNIM	83.4	83.6
DNIM + $\text{NFA}_{\mathcal{N}}$	**84.9**	**91.2**

Table 4.4: Transfer learning from SIRST to IRSTD-850.

c) Robustness analysis In this subsection, we assess the robustness of our method compared to the baseline DNIM in two scenarios: weak training conditions and generalisation to new or noisy data.

Few-shot learning - In many real world applications, data collection and annotation requires expertise, which is very expensive and time consuming. Having a method that leads to good performance even with little training data is essential in such real world applications. In the subsection, we evaluate the robustness of our method in few-shot settings, by training the NN on 15 and 25 images from NUAA-SIRST dataset (representing respectively about 5% and 10% of the training set used in Section 4.2.3). DNIM and DNIM+$\text{NFA}_{\mathcal{N}}$ are trained on three different non-overlapping sets of data in both 15-shot and 25-shot settings, and the averaged results are given in Table 4.3. It can be seen that our method performs significantly better in a frugal setting than the baseline. Indeed, both AP and F1 metrics are increased by more than 15% when adding the NFA module to DNIM in a 15-shot training. Moreover, the AP is decreased by only 5.3% for DNIM+$\text{NFA}_{\mathcal{N}}$ (compared to 13.6% for the baseline) when dividing by 10 the number of training samples. The robustness of the NFA module towards frugal setting is explained by the *a contrario* paradigm introduced in the training loop: we force the NN to model the background elements (rather than the targets themselves), for which we have sufficient samples even in a few-shot setting.

Generalisation to noisy and new data - One essential property of strong detectors is their ability to correctly generalize to unseen data. To this end, we first evaluate the robustness of DNIM+$\text{NFA}_{\mathcal{N}}$ towards noisy data during the inference. We consider two types of noise: additive and multiplicative Gaussian noises, with different variances (namely 0.01, 0.05 and 0.10). For the additive Gaussian noise the mean is set to 0 while for the multiplicative one it is set to 1. As we can see



(a) Additive Gaussian noise

(b) Multiplicative Gaussian noise

Figure 4.8: Sensitivity of DNIM and DNIM+NFA$_{\mathcal{N}}$ towards noisy images from NUAA-SIRST during inference.

from Figure 4.8, although F1 score and AP decrease with the increase in variance for both methods, DNIM+NFA$_{\mathcal{N}}$ still achieves the best performance by a large margin, for both considered types of noise. It is also significantly robust towards false alarms (AP criterion) compared to DNIM, especially in the case of additive Gaussian noise (Figure 4.8a).

We finally evaluate the methods on new scenes by transferring the knowledge learned on NUAA-SIRST dataset to IRSTD-850 dataset, without fine-tuning. The results in Table 4.4 confirm the generalisation ability of our method on new challenging scenes. Compared to the baseline, the F1 score is increased by 1.5%, and the AP by 7.6%. This robustness to new or noisy data is explained by the use of a naive model, which can only be approximate (provided that it contradicts the detections).

Σ **(Eq.**(4.1)**)**	**MS**	**ECA**	**Smooth**	**F1**	**AP**
$\Sigma = \lambda I_d$			✓	$96.0^{\pm 0.9}$	$97.6^{\pm 0.9}$
Dense Σ			✓	$95.1^{\pm 1.3}$	$96.3^{\pm 1.0}$
$\Sigma = \lambda\Delta$			✓	$96.9^{\pm 0.5}$	$\mathbf{98.6}^{\pm 1.0}$
$\Sigma = \lambda\Delta$	✓			$\underline{97.2}^{\pm 0.6}$	$95.6^{\pm 2.3}$
$\Sigma = \lambda\Delta$	✓		✓	$\underline{97.2}^{\pm 0.6}$	$97.9^{\pm 0.9}$
$\Sigma = \lambda\Delta$	✓	✓	✓	$\mathbf{97.6}^{\pm 0.3}$	$\underline{98.4}^{\pm 0.6}$

Table 4.5: Ablation study performed on NUAA-SIRST. We evaluated (object-level metrics) the different forms of the covariance matrix Σ and compared the benefits of multi-scaling (MS), adding a smoothing term (Smooth) and using channel attention (ECA) in our NFA module.

d) Ablation study Tables 4.5, 4.6 and 4.7 present the ablation and sensitivity studies performed on small target detection on NUAA-SIRST dataset. The conclusions are summarised in the five following points.

i) Assumptions made on the covariance distribution - In Section 4.1.1, we present three different forms for the covariance distribution Σ in Eq. (4.1), corresponding to three different assumptions about feature noise: spherical distribution, elliptical distribution with independent channels or components, elliptical distribution with dependent channels. The first two lines of Table 4.5 show that assuming a spherical distribution assumption leads to worse results, as does the channel-dependence assumption. To explain this, one has to remind that in deep learning, in order to disentangle causal factors, a series of filters are applied to extract the relevant characteristics. Each filter extracts a particular feature represented by a channel in the next feature maps. Depending on the downstream task, some features will be more or less relevant. The relevant information is therefore not equally distributed over all the channels of the feature maps. Besides, estimating the full covariance matrix in high dimensionality may be numerically unstable, while the correlation between extracted features remains low. As a result, the independent elliptical distribution appears as the more relevant hypothesis.

ii) Adding a smoothing term prevents from object fragmentation - According to Table 4.5, adding a spatial smoothing term, which is defined as the L2 norm of the gradient of the image in both vertical and horizontal directions, improves AP criterion. Indeed, it allows us to force low value difference between neighbouring pixels, and thus to avoid object fragmentation when increasing the segmentation threshold. The achieved results are then more robust to this threshold, which increases the AP.

iii) Importance of multiscaling - Adding information from low-level features helps the network in detecting objects of various size. This is the case for the baseline DNIM, whose multiscale version is DNANet, and it has been confirmed when adding NFA layers to the five scales of DNIM and considering the F1 criterion in Table 4.5. However, F1 increase is at the cost of a decrease of the AP criterion since introducing low-scale features may bring out more false positives for lower thresholds. We also performed an ablation study on the number of scales used in the NFA fusion block. We varied the parameter m in Eq. (4.4) from 2 to 5, 5 being the maximum number of scales in DNIM. The results presented in Table 4.6 show the importance of considering all the five scales, even for detecting small targets. Indeed, the F1 score increases gradually from 96.0% to 97.6% as more scales are added, and the AP reaches its maximum when considering five scales. The standard deviation also decreases when adding more scales, meaning that the NN gains in stability.

iv) Channel attention highlights the importance of high-level scales

for small target detection - To tackle previous issue, we introduced a channel attention layer before merging the different scales, that is, ECA block. Table 4.5 clearly shows the superiority of the NFA module when adding this step. It noticeably improves the average precision as well as the F1 score, by reducing the object false alarm rate. Looking at the multiplying factors computed by this channel attention layer, we observe that, for small target detection, the high-level features are of primary importance: their weight is about 0.99 when the weight of lower-level feature maps is about 20 times less, though they still contribute to the decision.

m (Eq.(4.4))	**F1**	**AP**
2	$96.0^{\pm 0.8}$	$96.7^{\pm 2.0}$
3	$96.5^{\pm 0.8}$	$98.3^{\pm 0.7}$
4	$97.6^{\pm 0.3}$	$97.8^{\pm 1.0}$
5	$97.6^{\pm 0.3}$	$98.4^{\pm 0.6}$

Table 4.6: Ablation study on the number of scales m in Eq.(4.4).

Activation function	**F1**	**AP**
Sigmoid	$90.5^{\pm 6.3}$	$93.5^{\pm 1.2}$
$\text{SIGM}_{\alpha=0.0001}$	$96.4^{\pm 1.4}$	$95.1^{\pm 0.1}$
$\text{SIGM}_{\alpha=0.0005}$	$\mathbf{97.2}^{\pm 0.6}$	$\mathbf{97.9}^{\pm 0.9}$
$\text{SIGM}_{\alpha=0.001}$	$96.7^{\pm 0.2}$	$94.8^{\pm 1.0}$

Table 4.7: Sensitivity study made on the activation function. Metrics are given at object-level.

v) Appropriate activation function - To confirm that conventional symmetric activation functions such as the sigmoid function are not suitable for the *significance* values, Table 4.7 shows the result obtained considering the sigmoid activation function, which indeed severely degrades the F1 score and AP of our method. The results are also less stable across different weight initialisations, as shown by the large standard deviation values (more than 6% in F1 score).

Now, for the proposed activation SIGM_α, as discussed in Section 4.1.2, the choice of α has an impact on the range of optimal thresholds for score map binarisation. We tested three values of α, which moves the upper bound for thresholds from 0.02 to 0.3. According to Table 4.7, $\alpha = 0.0005$ leads to the best performance, and we recommend to use this value.

4.3 Extension to other applications

We have shown in previous section that the NFA module can improve the performance of a segmentation NN specifically designed for small target detection. This allowed us to obtain state-of-the-art results on an application that represents an ideal framework for small object detection. Now, we propose to expand the boundaries of previous framework. For this purpose, we integrate our method in a classical semantic segmentation backbone and we apply it to two other applications, namely road crack detection and ship detection from remote sensing data. Both applications deal with small object detection in a frugal setting, and they are challenging for several reasons. In the case of road crack detection, the difficulty lies in the fact that i) the cracks are very thin and their pixels are very few with respect to the background class, and ii) the textured background and road artefacts can lead to numerous false alarms. Some generic deep learning approaches have been tested on this application, and are mainly based on classical segmentation NN [87, 88, 89]. Ship detection from low resolution satellite imagery is even more challenging because of i) the large number of boats in the same area and their varying sizes (e.g., pleasure boats, cargo ships), ii) the moored or tiny ships that are very hard to distinguish from decks or water wings, and iii) the low resolution of satellite data, which requires to use subpixel information. Most efficient methods for ship detection rely on data fusion (e.g., using SAR data, or the information provided by Automatic Identification Systems (AIS) [90]). Few deep learning methods for detecting ships from optical data have been proposed, and these detectors mainly rely on classical segmentation NN [11].

4.3.1 Assessed methods

We take as a baseline classical segmentation backbone, namely a UNet with a ResNet encoder (ResUNet, [85]). Note that, for crack detection, geometric information is crucial since the cracks exhibit a specific shape. Therefore, we take the opportunity given by crack detection to evaluate the contribution of the spatial NFA block in the NFA module. Based on the ablation study conducted in Section 4.2.3, we use the multi-scale NFA module with $\Sigma = \lambda\Delta$ (Eq. (4.1)) and set the parameter α in Eq. (4.5) to 0.0005. For both crack and ship detection, the theoretical threshold can be defined as follows. In the case of an image without cracks, one false alarm at a pixel level will not be significant for the application: indeed, a crack is defined by several hundred pixels. The same reasoning can be applied for boat detection as in the considered dataset there are many ships, including cargo ships that have a spatial extent of several hundred pixels. It is therefore reasonable to tolerate one pixel false alarm per image, which makes the false alarm expectation $\epsilon = 1$, leading to a binarization threshold $t \approx 0.001$. For

Method	CrackTree		S2SHIPS	
	F1	**AP**	**F1**	**AP**
ResUNet	$85.6^{\pm 0.4}$	$85.2^{\pm 0.2}$	$23.7^{\pm 2.0}$	$52.3^{\pm 6.4}$
ResUNet + $\text{NFA}_{\mathcal{N}}$	$\mathbf{87.2}^{\pm 0.0}$	$\mathbf{96.7}^{\pm 0.2}$	$\mathbf{35.3}^{\pm 1.5}$	$\mathbf{62.3}^{\pm 7.9}$

Table 4.8: Comparison of ResUNet and ResUNet+$\text{NFA}_{\mathcal{N}}$ on crack and ship detection. Metrics are computed at pixel-level for crack detection, and at object-level for ship detection.

a fair comparison with the baseline, whose optimal threshold no longer seems to be 0.5, we choose the threshold for the baseline based on the validation dataset. Both methods are trained for 700 epochs using the same loss and optimizer as in Section 4.2.2. ResUNet is trained with a learning rate of 0.01, and we lower the learning rate for ResUNet+$\text{NFA}_{\mathcal{N}}$ to 0.005.

4.3.2 Datasets and evaluation metrics

Crack detection - We train and evaluate all methods on CrackTree dataset from [91]. It is composed of 206 real pavement images, and it includes various types of cracks. Because very few data is available, the algorithms are trained using 120 images only. This frugal setting adds some challenge to the application. Finally, 36 images are used for the validation step, and 50 for testing. All methods are evaluated using pixel-level metrics, namely F1 score and average precision. However, as stated in [91], the annotations do not accurately report crack thickness. Therefore, like in the original paper, we adopt a tolerance margin of two pixels in crack localisation.

Ship detection - We consider the dataset S2SHIPS [11], which is composed of 16 multispectral images from Sentinel2 satellite sensor, of size 1783×938 pixels. Four images are kept for test, and the others are used for the train and validation datasets. From each image we extract 18 patches of size 256×256, which makes a total amount of 216 patches for the training and validation sets. We use the following six spectral channels as in [11]: B2 (B), B3 (G), B4 (R), B8 (NIR), B11 and B12 (SWIR). Note that in this application, training conditions are particularly difficult: there is very little training data, much of which does not include ships. The assessed methods are evaluated using F1 score and average precision computed at object-level.

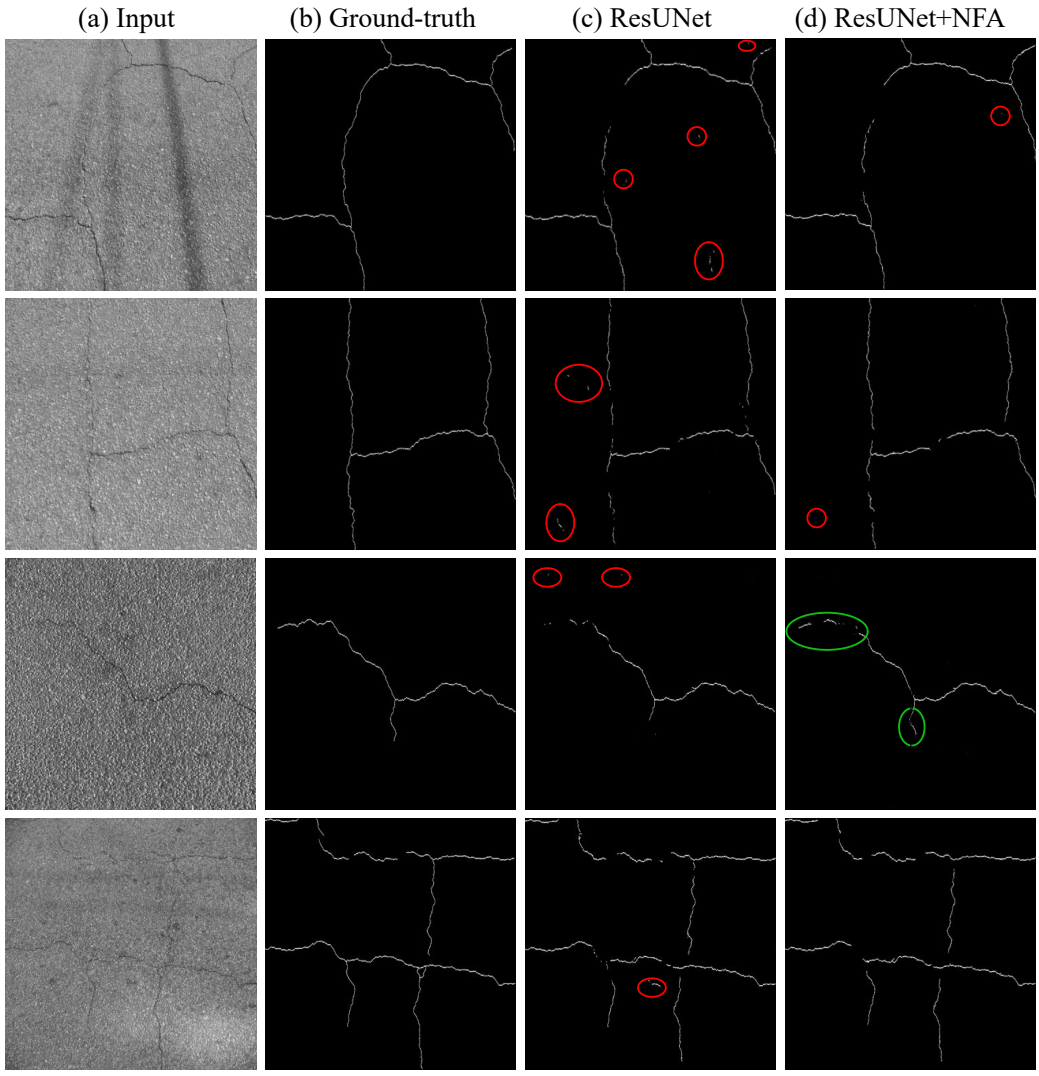


Figure 4.9: Qualitative results obtained with different detection methods on CrackTree dataset. False positives are circled in red, and reconstruction improvements are circled in green.

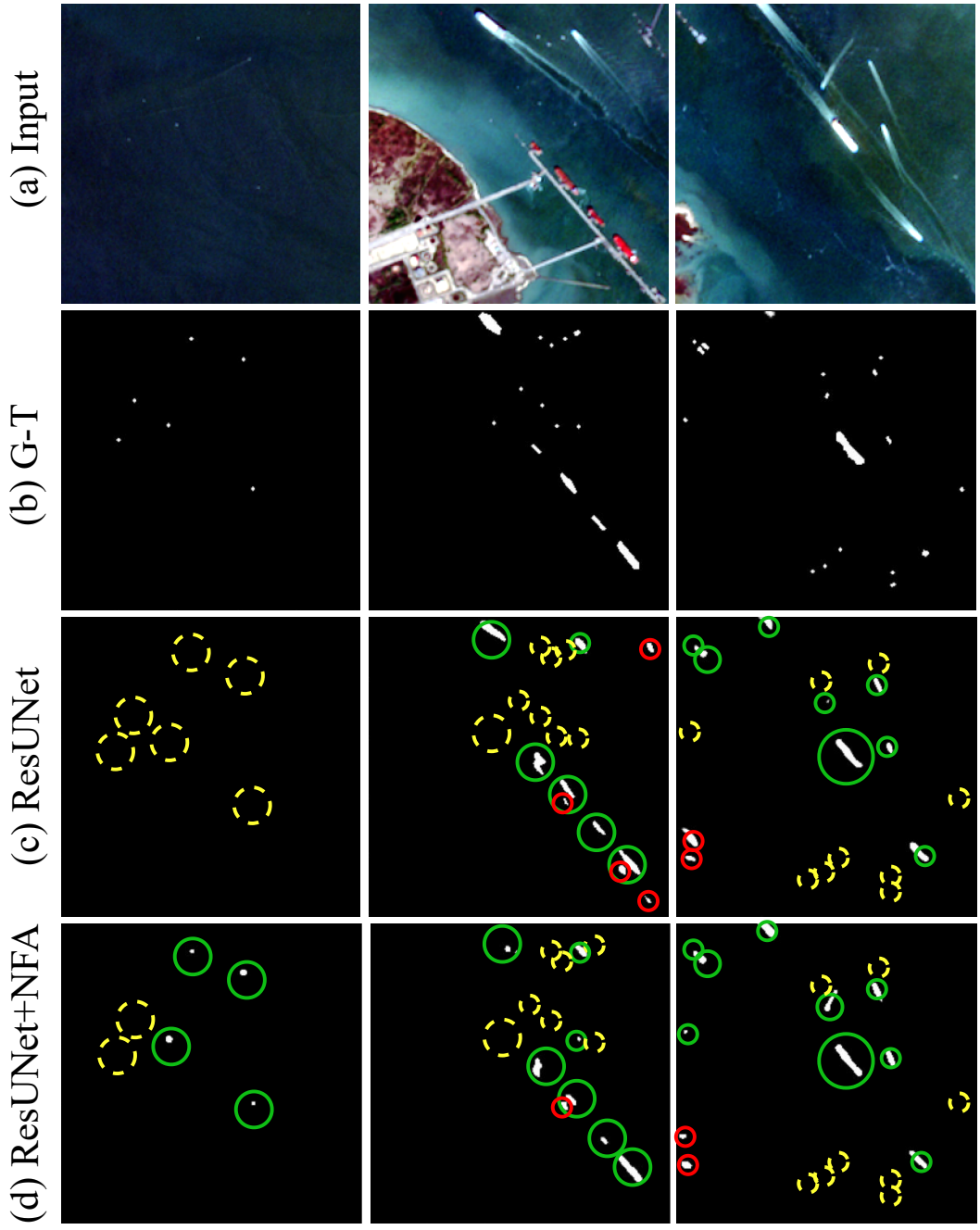


Figure 4.10: Qualitative results obtained with ResUNet and ResUNet+NFA$_{\mathcal{N}}$ on S2SHIPS dataset. True positives, false positives and missed detections are circled in green, red and dotted yellow lines, respectively.

4.3.3 Results

The NFA module leads to better performance

Table 4.8 shows the performance of the evaluated methods on CrackTree and S2SHIPS datasets. It is clear that the NFA module contributes in improving the baseline. Indeed, in the case of crack detection, the F1 score is increased by 1.4% when including the NFA module in the baseline. More precisely, we observe a very significant improvement in the average precision (more than 10%), which confirms the ability of the NFA module to control the number of false alarms at a pixel level. We notice that the recall is also improved. Figure 4.9 shows some results obtained by the different methods on four different crack examples. The NFA module appears more robust to the presence of shadows or textures on the road, since less false alarms are observed.

The same conclusions can be drawn for ship detection: both F1 score and AP are increased by at least 10%. However, despite a significant improvement of the baseline thanks to the NFA module, the performance remains weak: this is explained by the challenging conditions described in Section 4.3.2 and illustrated on the first row of Figure 4.10. Indeed, the presence of decks, coastlines, or even ship wakes leads to several false alarms. Nonetheless, even though both algorithms struggle to detect the most tiny ships, ResUNet+NFA$_{\mathcal{N}}$ considerably increases their detection as it can be seen on the first image. These experiments on two different applications confirm once more the robustness towards challenging conditions brought by our NFA module.

Contribution of attention mechanisms

We have evaluated the contribution of the different attention mechanisms, spatial attention and channel one separately, on crack detection. The results are detailed in Table 4.9 and our conclusions are summarised in the following points.

ECA	SASA	F1	AP
		$86.4^{\pm 0.1}$	$95.8^{\pm 0.2}$
✓		$\underline{87.0}^{\pm 0.3}$	$96.4^{\pm 0.1}$
	✓	$\underline{87.0}^{\pm 0.3}$	$\mathbf{96.8}^{\pm 0.2}$
✓	✓	$\mathbf{87.2}^{\pm 0.0}$	$\underline{96.7}^{\pm 0.2}$

Table 4.9: Ablation study performed on Crack Tree dataset (pixel-level metrics).

i) All scales are equally important for crack detection - In crack detection application, unlike in small target one where the decision process mainly relies on the high-level feature map (cf. Section 4.2.3), the deeper level feature

maps almost equally contribute to the prediction (multiplying factors all around 0.6). Indeed, the low resolution feature maps contain some useful information to describe large objects, while the high-level feature maps are meant for capturing the smaller details as outlined in [46].

ii) Spatial attention has a very significant impact on performance - As expected, the spatial attention block (SASA block) helps detecting precisely large objects. Indeed, thanks to spatial attention, the average precision is considerably improved: the shape of the cracks is estimated in an accurate way while eliminating some false positives.

Finally, combining both spatial and channel attention leads to even better and more stable results.

4.4 Conclusion

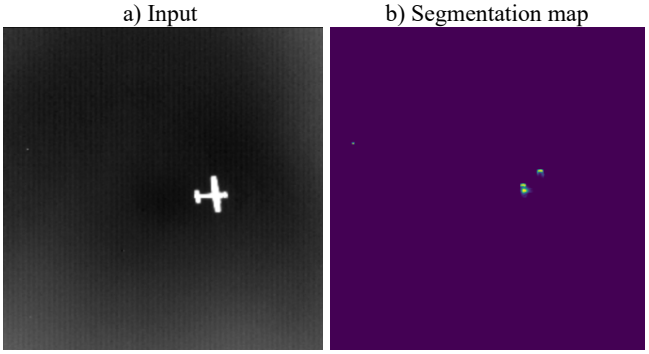


Figure 4.11: Behavior of DNIM+NFA$_{\mathcal{N}}$ on large objects.

Sections 4.2 and 4.3 show experimentally the benefits of our NFA module for tiny object detection. It significantly improves the performance of a conventional segmentation backbone such as ResUNet in three challenging applications, namely small target detection in infrared images, road crack detection in usual RGB images, and detection of ships from multispectral satellite images. Furthermore, we have shown that, when our NFA module is added on top of a backbone specifically designed for small target detection, such as DNIM, it is very competitive with the SOTA NN for small target detection DNANet, while being more interpretable. This improvement is due to the introduction of the *a contrario* paradigm in the training loop. The NN is forced to learn an approximate background model rather than the objects to be detected, by computing a number of false alarms (NFA). This gives new properties to the NN that includes such *a contrario* criterion: i) the control of the number of false alarms, which translates into a clear improvement

of the AP criterion, and ii) the ability to learn from few samples of the object to be detected. The latter property increases the robustness of the NN to frugal learning and helps to better generalise to unseen data, as shown experimentally in Section 4.2.3.

The use of our method can be extended to other small object detection tasks such as the early medical diagnosis, early forest fire detection, or the detection of buildings in rural areas. Nevertheless, our method was specifically designed for the detection of very small objects (with respect to the number of image pixels), and we cannot expect good performance on large object detection. Indeed, as the proportion of the objects in the image increases, not only does the NN struggle to separate the distributions of the background from those of the objects, but also the unexpectedness feature of the targets becomes less pregnant. One consequence will be a fragmented detection of large objects, as illustrated in Figure 4.11. Only the 'hottest' points of the aircraft will show up on the segmentation map. Therefore, there will be three detections for a single object, which will artificially increase the number of false alarms. In the next chapter, we propose to integrate an *a contrario* criterion into object detection method in order to bypass this issue.

Chapter 5

Integration within object detection methods

In the previous chapter, we have presented the integration of a NFA (through a NFA module that computes $\text{NFA}_{\mathcal{N}}$) into segmentation networks, which are known to be SOTA for IRSTD. Such a module improves the performance of several segmentation networks, and for several applications. However, segmentation networks are not optimal for object detection, since an additional post-processing step is needed in order to perform object-level detection; indeed, without this step, they are also prone to object fragmentation. One possible solution is to use of object detection methods, such as Faster R-CNN or YOLO algorithms. The latter are widely used in object detection tasks, having proved to be highly effective in a variety of applications. However, YOLO networks lead to particularly weak performance when it comes to detecting small targets. Indeed, if the object to detect is too small, it may occupy only a small portion of a grid cell, making it difficult for YOLO to detect it accurately. To address this issue, YOLOv3 [48] introduced a feature pyramid network (FPN) that combines the features detected at multiple scales. This helps in enhancing small object features, and YOLO versions from YOLOv3 onwards lead to SOTA results on several object detection benchmarks. However, the performance remains beyond SOTA segmentation methods for small target detection. We try to go further by introducing an *a contrario* criterion into the training loop of an object detection network. More specifically, we propose to adapt two NFA formulations (based on two different naive models) for YOLO backbones. The NFA test will be used to re-estimate the objectness scores predicted by a YOLO network. In the following, we first introduce the different NFA detection heads as well as their integration within the YOLO framework. Then, we evaluate the different NFA heads on IRSTD. More specifically, we provide the results obtained on SIRST an IRSTD-850 datasets, then we evaluate the robustness of our methods towards frugal training, and finally we extend their application to

vehicle detection from remote sensing data.

5.1 Methodology

In this section, we present three NFA detection heads adapted for YOLO architectures: one pixel-level formulation based on $\text{NFA}_{\mathcal{N}}$ introduced in Chapter 4, and two object-level versions that exploit the spatial context given by the bounding boxes predicted by YOLO network to compute the *significance* score. For the latter two, our first motivation is to explicitly take into account both the spatial context and its grey level characteristics for the tested grid-cell (related to the objectness score) in the computation of the NFA. The second motivation is that an object-level approach seems more relevant for a detector based on box proposal such as YOLO. Indeed, the pixel-level formulation computes the *significance* score by considering exclusively one grey-level value of the objectness score (representing the center of a potential detected object), and thus does not take advantage of the spatial extent of the tested object provided by the bounding box proposals of YOLO. Furthermore, assessing two different NFA formulations (object-level and pixel-level) will give a more complete insight on the strengths and limitations of the *a contrario* paradigm for detecting small targets. Note that all the proposed NFA detection heads can be used to replace the detection head of any YOLO network.

5.1.1 Pixel-level $\text{NFA}_{\mathcal{N}}$ for object detection

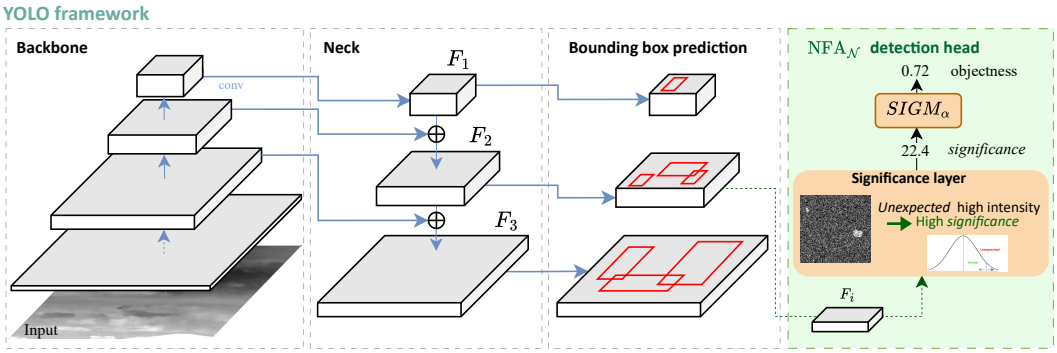


Figure 5.1: Integration of our pixel-level criterion into a YOLO framework, through the $\text{NFA}_{\mathcal{N}}$ detection head. This module can be added on top of any YOLO.

We first propose to integrate, within YOLO framework, the version of the NFA designed for segmentation networks in Section 4.1.2. As a reminder, the $\text{NFA}_{\mathcal{N}}$ is defined at pixel-level as follows:

$$\text{NFA}_{\mathcal{N}}(x_i, N_{test}, K, \Sigma) = \frac{N_{test}}{\Gamma(K/2)}\Gamma\left(\frac{K}{2}, \frac{1}{2}||\Sigma^{-1/2}x_i||_2^2\right),$$

where $\Gamma(.)$ and $\Gamma(.,.)$ are the Gamma and upper incomplete Gamma functions respectively, and where Σ represents the covariance matrix of the centred variable X_i. Based on the results obtained in Chapter 4, Σ is defined as $\lambda\Delta$ where Δ is a diagonal matrix with $|\Delta| = 1$ and $\lambda \in \mathbb{R}^{*+}$.

The integration of the $\text{NFA}_{\mathcal{N}}$ criterion, and more specifically the computation of the *significance* (cf Eq. (4.2)), is illustrated on Figure 5.1. To do so, we first modify the YOLO detection head by separately predicting the bounding box coordinates, the classification scores and the objectness scores. We then replace the original convolution step for predicting the objectness score by the basic NFA block (which computes Eq. (4.1), cf. Figure 4.3a). Finally, we apply the SIGM_{α} activation function to obtain an objectness score that ranges between 0 and 1. Note that since the YOLO version we consider is multi-scale, we obtain several *significance* maps. As specified in Section 4.1.2 of Chapter 4, each scale has the same weight in the decision, since we define a constant value for η_{test}. This strategy may be suboptimal in some cases, where many large objects need to be detected. For this purpose, we introduce some weighting coefficients, which are obtained using an attention layer as in Section 4.1.2. The integration of an ECA layer is omitted in Figure 5.1 for simplicity, but is similar to its integration into segmentation networks (cf. Figure 4.2).

We train the YOLO+$\text{NFA}_{\mathcal{N}}$ in an end-to-end manner using the Mean Squared Error loss as it has shown to lead to better performance in our experiments.

5.1.2 Object-level NFA: first version

One drawback of the previous formulation is that it provides an objectness score for each grid-cell (which represents the centre of the object, and has a limited spatial extent) and does not take into account the size of the predicted bounding box. We propose to explore and adapt an object-level formulation of the NFA to a YOLO architecture, in order to explicitly take into account the spatial extent of the predicted bounding box.

NFA formulation – Let us first describe our object-level NFA formulation for a single-channel feature map. It is similar to the formulation we introduced in Section 3.4.2, where we simultaneously consider grey level characteristics and spatial

structuring (point density). As a reminder, we consider that a set of pixels likely to represent a target is all the more significant as it contains many points that are spatially close and of high value on the score map. In contrast, we assume in our first object-level NFA version that the points are uniformly distributed within a bounded 2D space rather than a 3D space. Therefore, the naive model is the Bernoulli distribution of parameter p representing the presence of a pixel at a given position in a discrete and bounded 2D space $\mathcal{E} \subset \mathbb{R}^2$, with the axes representing the spatial coordinates. The probability of observing at least κ pixels in a rectangle of area ν is then the Binomial distribution of parameter p. Therefore,

$$\mathrm{NFA}_{\mathcal{U}_1}(\kappa, \nu, p) = \eta_{test} \sum_{i=\kappa}^{\nu} \binom{\nu}{i} p^i (1-p)^{\nu-i}, \tag{5.1}$$

where η_{test} is the number of tests, i.e. the number of tested 2D tiles (or cuboids for the second version of object-level NFA, cf. Section 5.1.3).

The previous formulation can only be applied to single-channel feature maps. We adapt this formulation to multi-channel feature maps by considering a 3D space instead of a 2D space, where the third axis represents the channels of the feature map. More specifically, we compute the number of points κ in the volume ν, and compare it to the 3D density p of the feature map using Eq. (5.1).

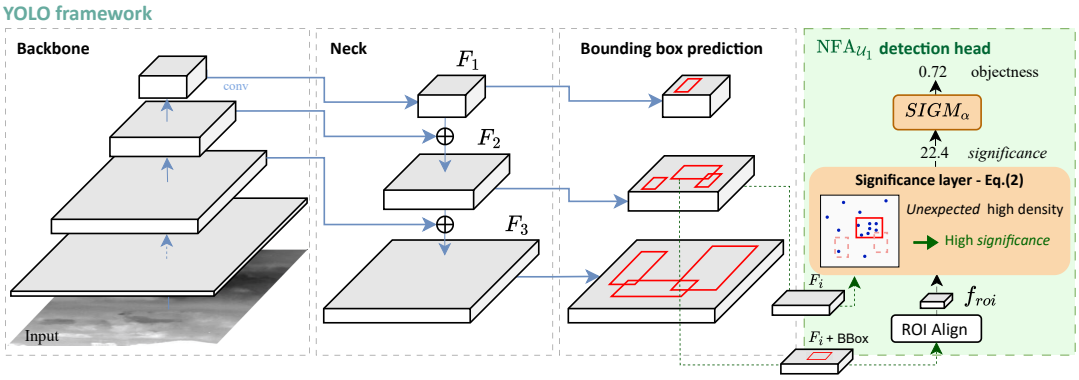


Figure 5.2: Integration of our object-level criterion into a YOLO framework, through the $\mathrm{NFA}_{\mathcal{U}_1}$ detection head. This module can be added on top of any YOLO.

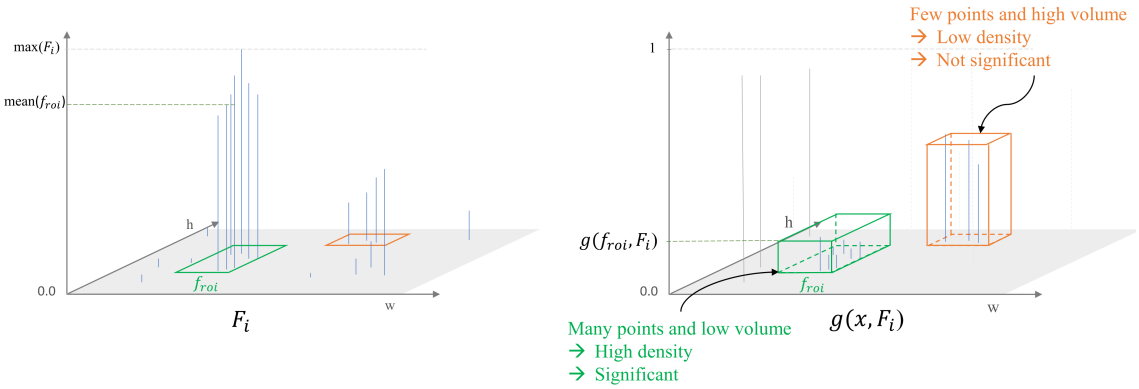
Integration within YOLO – The overall architecture of our approach is illustrated in Figure 5.2. The infrared input images first go through a YOLO backbone that extracts the feature maps at different scales. Then, the three lower-level features are combined together through the neck, which gives us the final feature

maps $\{F_i\}_{i\in[\![1,\dots,3]\!]}$ used to perform the detection at three levels. To achieve the detection, the bounding box coordinates are first predicted through a dense layer. We then introduce our $\text{NFA}_{\mathcal{U}_1}$ module to re-estimate the objectness score for each bounding box using the NFA criterion. To do so, we extract η_{test} ROI, denoted as f_{roi}, using ROI Align from Faster R-CNN [45], and we compute a *significance* score $S_{\mathcal{U}_1}(\kappa,\nu,p) = -\ln(\text{NFA}_{\mathcal{U}_1}(\kappa,\nu,p))$ for each ROI through the significance layer. Finally, since the *significance* values range from $[-\ln(\eta_{test}),+\infty)$, where large values correspond to possible targets, to obtain objectness scores that range in $[0,1]$, we apply the SIGM_α activation function with $\alpha = 0.5$. This allows us to apply the Binary Cross Entropy loss used in YOLO. As in Section 5.1.1, we weight each spatial scale by applying a channel-attention layer.

Our significance layer in Figure 5.2 integrates the *a contrario* criterion given in Eq. (5.1). However, since this equation is (i) designed for binary images rather than greyscale feature maps, and (ii) not differentiable, several approximations were made in order to allow its integration into the YOLO training loop. The first difficulty raised by Eq. (5.1) is to count the number of "true" pixels κ in $f_{roi} \in \mathbb{R}^2$. Thresholding f_{roi} to binarise it would break the back-propagation loop. Thus, we propose instead to consider real number membership coefficients (in the spirit of fuzzy clustering or classification), which boils down to handling, for each pixel, a coefficient indicating the degree to which it belongs to the set containing pixels with a value of 1 in the binary case. For this purpose, we apply the sigmoid function σ on the pixel values, which allows us to approximate the number of pixels contained in f_{roi} for estimating the local density, by the sum of these fuzzy belonging coefficients. The same approximation is made to compute the total number of points in F_i for estimating the parameter p (representing the global density of F_i) of the binomial law in Eq. (5.1). The second issue is that the $\text{NFA}_{\mathcal{U}_1}$ function is discontinuous, non differentiable and, as we deal with objects having a small area ν, it only takes very few distinct values. These elements make it difficult to integrate Eq. (5.1) "as is" into the training loop, with a working back-propagation. We therefore use the Hoeffding approximation when $\frac{\kappa}{\nu} > p$ for computing the *significance*, leading to

$$S_{\mathcal{U}_1}(\kappa,\nu,p) \approx \nu\left[\frac{\kappa}{\nu}\ln\left(\frac{\frac{\kappa}{\nu}}{p}\right) + \left(1-\frac{\kappa}{\nu}\right)\ln\left(\frac{1-\frac{\kappa}{\nu}}{1-p}\right)\right] - \ln\eta_{test}. \tag{5.2}$$

This allows us to expand the codomain of the function $S_{\mathcal{U}_1}(\kappa,\nu,p)$ to $\mathbb{R}$, and to output more intermediate values. In the case of $\frac{\kappa}{\nu} \leq p$, we simply assign $S_{\mathcal{U}_1}(\kappa,\nu,p) = -\ln(\eta_{test})$ using ReLU activation function, as it corresponds to obvious background values.



(a) Feature map F_i

(b) F_i after applying the decreasing transform $g(x, F_i)$

Figure 5.3: Illustration of the $\text{NFA}_{\mathcal{U}_2}$ formulation. Sub-figure (a) represents the feature map F_i, and a third axis representing the transformed values $g(x, F_i)$ is introduced in sub-figure (b).

5.1.3 Object-level NFA: second version

Note that, with the object-level $\text{NFA}_{\mathcal{U}_1}$ formulation described in Section 5.1.2, applying the sigmoid function σ to approximate the number of pixels contained in f_{roi} reduces the dynamics of the feature maps, resulting in a loss of information. To counter this, we propose a 3D version of the previous object-level $\text{NFA}_{\mathcal{U}_1}$ formulation, where the third axis represents the transformed score values instead of the channels of the feature map. This second object-level formulation is coined as $\text{NFA}_{\mathcal{U}_2}$.

To achieve this, the score values are transformed so that the low score values lead to a high volume (and thus a less significant region), while high score values define a dense region (i.e., a cuboid with a low volume). More precisely, for computing ν in a 3D space where the spatial area is defined by f_{roi}, we estimate the height of the cuboid by computing the distance between the average value of f_{roi} and the maximum value of the feature maps. The same approach is used for computing p in the 3D space defined by F_i. For this purpose, the following decreasing function is introduced: $g(x, F_i) = \max(1, \max(F_i) - \text{mean}(x))$, where x is a region in the feature map F_i, $\max(\cdot)$ is the function that provides the maximum value of an element and $\text{mean}(\cdot)$ its average value. The function $g(x, F_i)$ ensures that its minimum height is 1, allowing to compute a proper volume. By using the functions $g(f_{roi}, F_i)$ and $g(F_i, F_i)$ for computing the volume ν defined by f_{roi} and the density p defined by F_i respectively, a tested box with high greyscale values will result in a high density box (i.e., many points close to 1 contained in a little volume), meaning that this box is *significant* with respect to the $\text{NFA}_{\mathcal{U}_2}$ criterion.

An illustration of the $\text{NFA}_{\mathcal{U}_2}$ formulation is given in Figure 5.3. In practice, to increase the robustness of the training process, we estimate $\max(F_i)$ globally across all training batches using exponential moving average.

The multi-channel version of the $\text{NFA}_{\mathcal{U}_2}$ formulation is slightly different since we cannot consider, at the same time, both the channels and the greyscale level as the third axis of the cuboid. Therefore, to take into account the multi-channel characteristic of the feature maps, we first compute the significance associated to $\text{NFA}_{\mathcal{U}_2}$ for each channel independently, and then we merge the obtained *significance* maps by taking the union of detections, as described in Section 4.1.2.

5.2 Experiments

Notation	Description
$\text{NFA}_{\mathcal{N}}$	Pixel-level NFA head based on Eq. (4.1).
$\text{NFA}_{\mathcal{U}_1}$	First version of the object-level NFA head introduced in Section 5.1.2. It is based on the 3D version of Eq. (5.1) where the third axis represents the channel-wise dimension of the feature maps.
$\text{NFA}_{\mathcal{U}_2}$	Second version of the object-level NFA head introduced in Section 5.1.3. It is also based on Eq. (5.1) but in this case the third axis represents a transformation of the feature map scores. The multi-channel aspect is taken into account by merging the NFA maps obtained independently for each channel (by taking the union of detections).

Table 5.1: Notations and description of the different NFA detection heads.

In this section, we evaluate the benefits of our different NFA heads for YOLO-based small object detection. More precisely, we evaluate three NFA formulations: the pixel-level $\text{NFA}_{\mathcal{N}}$, and the two-object-level versions $\text{NFA}_{\mathcal{U}_1}$ and $\text{NFA}_{\mathcal{U}_2}$. Table 5.1 recalls briefly the characteristics of each NFA formulation. We first add our NFA detection heads on the YOLOv7-tiny baseline, and compare our results to several baselines: 1) generic YOLO baselines[1] such as YOLOv3 [48], YOLOR [92], YOLOv7 and YOLOv7-tiny [49] and 2) SOTA segmentation networks introduced in Section 4. We also evaluate our methods in different scenarios. We first evaluate them on infrared small target detection tasks by considering NUAA-SIRST and IRSTD-850 datasets, with the train/val/test splits described in Section 4.2.2. As YOLO networks are designed to take large image inputs, we upsample all images to the size 640×640 using bicubic interpolation. Second, we study the

[1]For YOLO baselines, we used the official PyTorch implementation of YOLOWongKinYiu/yolov7.

robustness of our methods towards few-shot training and knowledge transfer from SIRST dataset to IRSTD-850 dataset. Then, we discuss incorporating our NFA head into small object friendly YOLO architectures, and finally, we propose to extend the application of our methods to the detection of slightly larger objects (but still considered as small objects) with the dataset VEDAI. All networks are trained from scratch on Nvidia RTX6000 GPU for 600 epochs, with Adam optimiser [93], a batch size equal to 16 and a learning rate equal to 0.001. We use the same data-augmentation functions as those proposed by default in YOLOv7-tiny implementation.

5.2.1 *A contrario* reasoning benefits YOLO-based IRSTD

Method	F1	AP	Prec.	Rec.	#params
SOTA segmentation networks for IRSTD					
DNANet [8]	$\underline{96.9}^{\pm 0.5}$	$\underline{98.1}^{\pm 1.2}$	96.6	97.2	4.7 M
DNIM [8] + NFA$_{\mathcal{N}}$	$\mathbf{97.6}^{\pm 0.3}$	$\mathbf{98.4}^{\pm 0.6}$	97.9	97.4	4.8 M
Object detection methods					
YOLOv3 [48]	$96.1^{\pm 0.3}$	$97.5^{\pm 0.1}$	96.9	95.4	61.5 M
YOLOR [92]	$95.7^{\pm 2.2}$	$96.7^{\pm 1.1}$	96.5	94.9	52.5 M
YOLOv7 [49]	$96.5^{\pm 1.2}$	$97.6^{\pm 0.7}$	97.2	95.9	36.9 M
YOLOv7-tiny	$96.5^{\pm 0.6}$	$97.8^{\pm 0.4}$	96.9	96.2	6.0 M
YOLOv7-tiny + NFA$_{\mathcal{N}}$	$\mathbf{97.6}^{\pm 0.3}$	$\mathbf{98.3}^{\pm 0.1}$	97.6	97.6	6.5 M
YOLOv7-tiny + NFA$_{\mathcal{U}_1}$	$96.8^{\pm 1.6}$	$\underline{98.1}^{\pm 0.3}$	99.6	94.2	6.4 M
YOLOv7-tiny + NFA$_{\mathcal{U}_2}$	$\underline{97.0}^{\pm 0.3}$	$\mathbf{98.3}^{\pm 0.3}$	98.6	95.5	6.4 M

Table 5.2: Object-level metrics (F1, AP, Prec., Rec.) achieved by the compared methods on NUAA-SIRST. Best results are in bold and second best results are underlined. The number of training parameters (#params) is also given.

Tables 5.2 and 5.3 provide the performance achieved by each of the compared methods on NUAA-SIRST and IRSTD-850 respectively. More specifically, the contribution of our three NFA head versions is compared to some YOLO baselines, as well as to SOTA segmentation networks for IRSTD, namely DNANet and DNIM+NFA$_{\mathcal{N}}$ (cf. Chapter 4).

Let us first analyse the results obtained on NUAA-SIRST, which is our reference dataset. From Table 5.2 we can see that substituting the conventional YOLO detection head with one of our NFA heads (NFA$_{\mathcal{N}}$, NFA$_{\mathcal{U}_1}$ or NFA$_{\mathcal{U}_2}$) benefits the detection of small targets of YOLOv7-tiny backbone. Indeed, the object-level NFA heads NFA$_{\mathcal{U}_1}$ and NFA$_{\mathcal{U}_2}$ increase the F1 value of the YOLOv7-tiny baseline

Method	F1	AP	Prec.	Rec.
SOTA segmentation networks for IRSTD				
DNANet [8]	$\mathbf{91.4}^{\pm 1.4}$	$92.4^{\pm 1.9}$	91.8	91.1
DNIM [8] + NFA$_{\mathcal{N}}$ [94]	$\underline{91.3}^{\pm 0.7}$	$\mathbf{94.2}^{\pm 0.2}$	92.1	90.6
Object detection methods				
YOLOv7-tiny	$84.0^{\pm 3.9}$	$88.8^{\pm 3.3}$	85.1	82.3
YOLOv7-tiny + NFA$_{\mathcal{N}}$	$90.1^{\pm 1.1}$	$\underline{94.1}^{\pm 0.5}$	91.8	90.7
YOLOv7-tiny + NFA$_{\mathcal{U}_1}$	$86.0^{\pm 2.3}$	$90.4^{\pm 0.9}$	88.2	84.3
YOLOv7-tiny + NFA$_{\mathcal{U}_2}$	$86.8^{\pm 1.5}$	$89.2^{\pm 2.2}$	85.8	87.9

Table 5.3: Object-level metrics (F1, AP, Prec., Rec.) achieved by the compared methods on IRSTD-850. Best results are in bold and second best results are underlined.

by at least 0.3%. The precision criterion is significantly improved, which shows that object-level NFA heads efficiently control the false alarm rate. Note that the NFA$_{\mathcal{U}_2}$ version seems to be more efficient and robust than the NFA$_{\mathcal{U}_1}$ version: the compression of the grayscale dynamics in the NFA$_{\mathcal{U}_1}$ version may be the reason for this observation.

However, our pixel-level NFA$_{\mathcal{N}}$ head performs significantly better than the other versions. Indeed, it increases the baseline F1 score by 1.1% and the AP criterion by 0.5%. It not only improves the precision but also the recall criterion, which is one of the main weakness of YOLO backbones. Indeed, the YOLO backbones struggle to detect the small objects, as explained in Section 5.1. Adding an *a contrario* criterion, more specifically the pixel-level version, helps in enhancing small object features and thus discriminating them from complex backgrounds. Our pixel-level NFA$_{\mathcal{N}}$ head not only improves YOLO for tiny object detection, but also bridges the performance gap observed between SOTA IRSTD segmentation NN and conventional object detection NN. Specifically, our method performs better in terms of F1 and AP criteria than DNANet (+0.7%), and is on par with the method we proposed in Chapter 4, namely DNIM+NFA$_{\mathcal{N}}$. In terms of computational complexity, our NFA heads add very few training parameters (less than 0.5 million), which allows us to consider real-time detection with YOLO backbones. Figure 5.4 shows some predictions of the baseline YOLOv7-tiny and our method. We can see that the baseline leads to several false alarms for inputs shown in columns 3 and 4, while our method provides correct detections without any false alarm.

Table 5.3 provides the results obtained on IRSTD-850 dataset. We can first notice that YOLO baseline performs significantly worse than SOTA segmentation networks. Indeed, YOLOv7-tiny achieves a F1 value of 84.0%, which is 6.4%

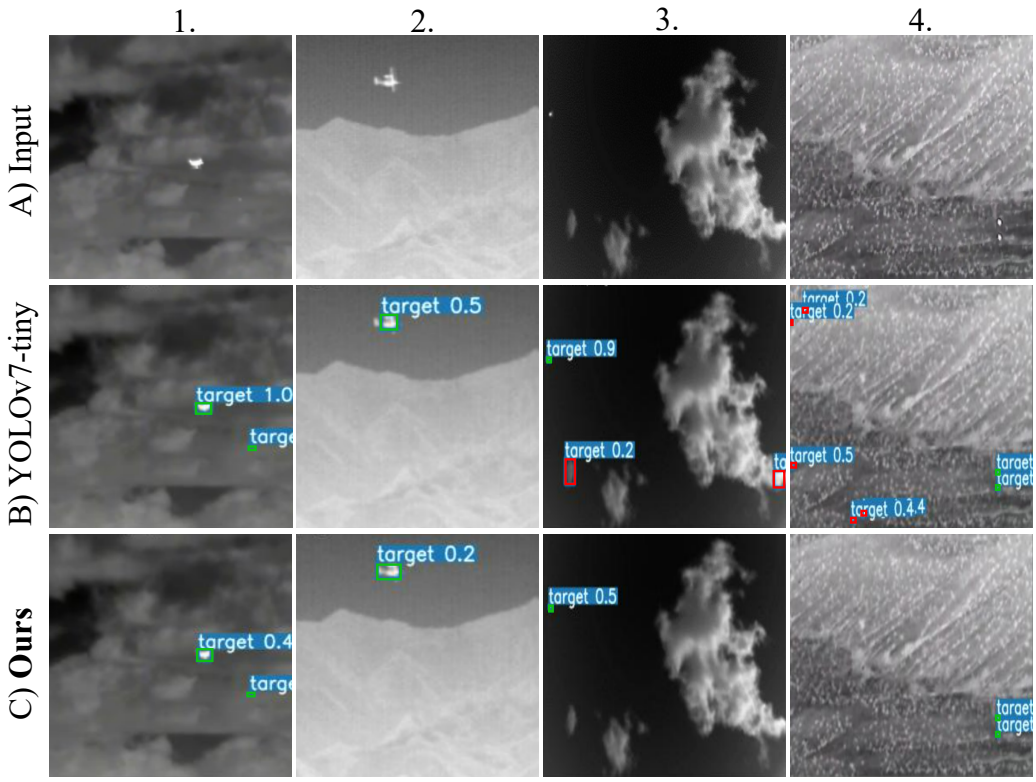
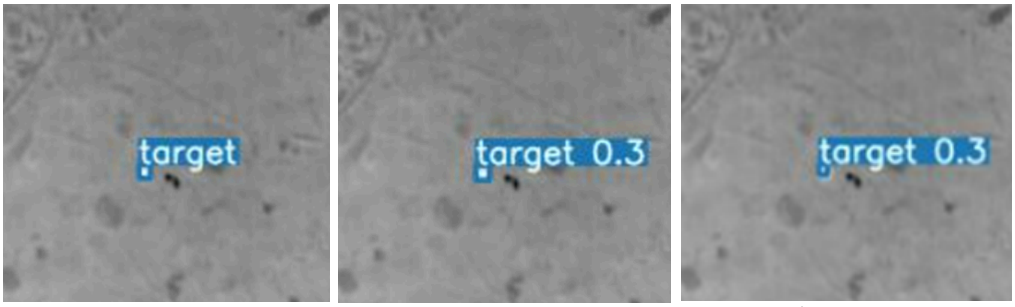


Figure 5.4: Qualitative results obtained with YOLOv7-tiny and our method (NFA head) on NUAA-SIRST dataset. Good detections and false positives are framed in green and red, respectively.

lower than DNANet. The recall rate is particularly low, which means that several targets are not detected. Unlike NUAA-SIRST dataset, IRSTD-850 contains more tiny targets, and it is a well-known fact that YOLO backbones struggle with tiny object detection. Adding the pixel-level $\text{NFA}_{\mathcal{N}}$ head significantly improves the performance, although it does not reach the F1 score of DNANet. Nonetheless, our $\text{NFA}_{\mathcal{N}}$ detection head bridges the performance gap between YOLO baselines and SOTA segmentation networks for IRSTD. Figure 5.6 show some predictions.

From Table 5.3, we can also notice that object-level NFA heads tend to perform worse than the pixel-level NFA head. A deeper analysis of the results allows us to hypothesise that this decrease in performance is mostly due to a box localisation error. Indeed, if we decrease the IoU constraint for evaluating whether a predicted box is a true positive or a false alarm, we can observe that the performance increases only for object-level NFA heads, not for the baseline or for the $\text{NFA}_{\mathcal{N}}$ head. More specifically, by fixing the IoU to 1% instead of 5%, YOLOv7-tiny+$\text{NFA}_{\mathcal{U}_1}$ leads to an F1 score of 87.8% and YOLOv7-tiny+$\text{NFA}_{\mathcal{U}_2}$ reaches a



a) Ground-truth b)$\text{NFA}_{\mathcal{N}}$ c)$\text{NFA}_{\mathcal{U}_2}$

Figure 5.5: Illustration of the bounding box localisation error for $\text{NFA}_{\mathcal{U}_2}$ detection head.

F1 score of 87.2%. As explained in Section 4.2.2, the evaluation of tiny object detection is particularly sensitive to the choice of the IoU, and a small error in the box localisation can have a large impact on the IoU with the ground-truth.

Figure 5.5 provides another argument in favour of our hypothesis: we can notice that the bounding boxes predicted by our object-level NFA heads are smaller than both the ground-truth and those predicted by the baseline. This could be a bias induced by the NFA formulation, which considers very dense boxes to be more significant. For a same object, the box inside the object is likely to be more dense in terms of the number of points with value 1 than the box surrounding the object. This may explain the box localisation error and thus the poor performance in detecting very tiny objects. Further experiments or strategies to overcome the problem are left for future work. In the next sections, we will consider only the pixel-level $\text{NFA}_{\mathcal{N}}$ and object-level $\text{NFA}_{\mathcal{U}_2}$ formulations for our experiments as they have shown to be more efficient than $\text{NFA}_{\mathcal{U}_1}$ version.

To conclude this subsection, we have seen that adding the *a contrario* paradigm into the training loop of a YOLO network improves the performance, especially when considering the pixel-level $\text{NFA}_{\mathcal{N}}$ head. More specifically, it both controls the number of false alarms and improves the recall rate, which is one major weakness of YOLO backbones when it comes to small object detection. An attempt of qualitative explanation for the control of the number of false alarm can be provided by Figure 5.7, which shows the objectness score map given by the highest-level scale for YOLOv7-tiny, YOLOv7-tiny+$\text{NFA}_{\mathcal{U}_1}$ and YOLOv7-tiny+$\text{NFA}_{\mathcal{N}}$. The objectness map obtained by YOLOv7-tiny is particularly noisy, although the target stands out against the noise. We can clearly see the contribution of the NFA layer, which efficiently removes the noise. This leads to a clean segmentation map and makes it easier to choose a threshold on the objectness score while guaranteeing a high precision value.

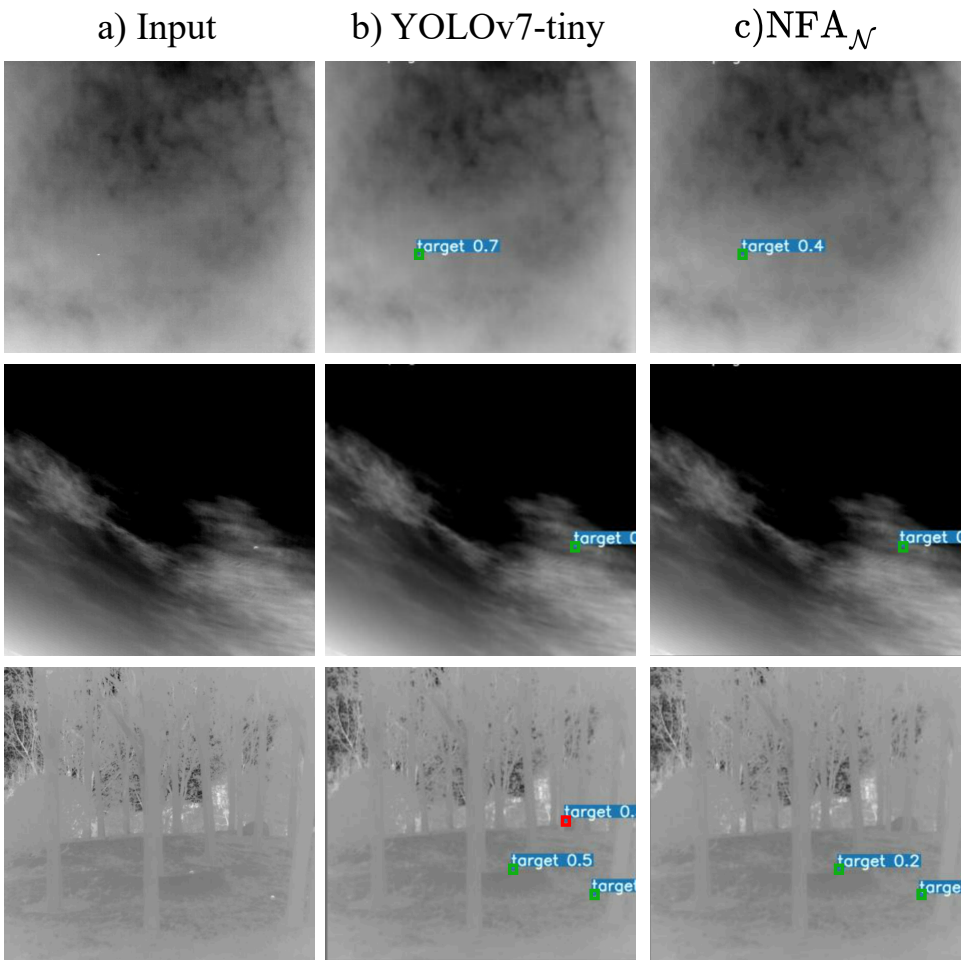


Figure 5.6: Qualitative results obtained with YOLOv7-tiny and YOLOv7-tiny + NFA$_{\mathcal{N}}$ on IRSTD-850 dataset. Good detections and false positives are framed in green and red, respectively.

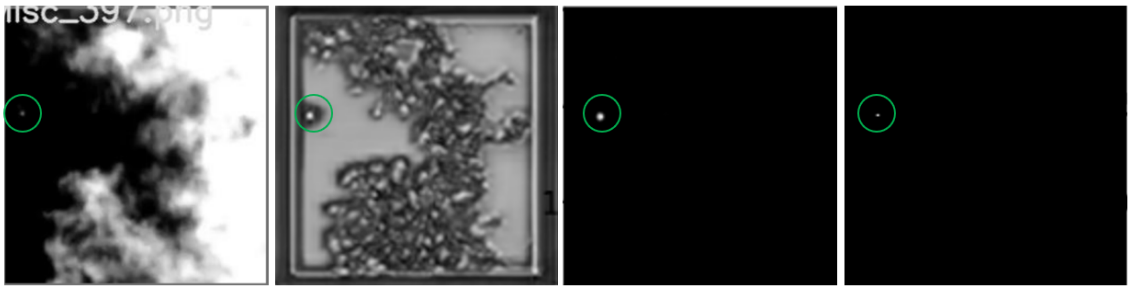


Figure 5.7: Objectness score feature maps. From left to right: original image and objectness score maps obtained by YOLOv7-tiny, YOLOv7-tiny + $\text{NFA}_{\mathcal{U}_1}$, YOLOv7-tiny + $\text{NFA}_{\mathcal{N}}$.

5.2.2 Robustness analysis

Method	25-shots		35-shots		45-shots	
	F1	**AP**	**F1**	**AP**	**F1**	**AP**
YOLOv7-tiny	$21.8^{\pm 13.6}$	$15.0^{\pm 13.4}$	$53.1^{\pm 21.4}$	$52.8^{\pm 26.9}$	$56.6^{\pm 7.7}$	$60.3^{\pm 9.9}$
+ **$\text{NFA}_{\mathcal{N}}$**	$\mathbf{93.6}^{\pm 1.2}$	$\mathbf{95.0}^{\pm 0.2}$	$\mathbf{94.8}^{\pm 0.9}$	$\mathbf{96.9}^{\pm 1.1}$	$\mathbf{95.5}^{\pm 0.3}$	$\mathbf{97.7}^{\pm 0.2}$
+ **$\text{NFA}_{\mathcal{U}_2}$**	$75.0^{\pm 6.9}$	$79.3^{\pm 7.4}$	$87.8^{\pm 1.0}$	$92.8^{\pm 0.6}$	$91.6^{\pm 1.9}$	$94.3^{\pm 1.4}$

Table 5.4: Results achieved in 25, 35 and 45-shot settings on NUAA-SIRST. Best results are in bold.

Method	F1	AP
DNIM	83.4	83.6
DNIM + $\text{NFA}_{\mathcal{N}}$	**84.9**	**91.2**
YOLOv7-tiny	74.4	77.4
YOLOv7-tiny + $\text{NFA}_{\mathcal{N}}$	79.8	82.5
YOLOv7-tiny + $\text{NFA}_{\mathcal{U}_2}$	75.4	78.4

Table 5.5: Knowledge transfer from NUAA-SIRST to IRSTD-850.

One important motivation for integrating *a contrario* reasoning into a NN is that the network learns to discriminate small targets by learning a representation of the background elements rather than the targets themselves. As seen previously, this leads to cleaner objectness score maps, which illustrates the effective control of the false alarms. It should thus provide robustness to the NN against weak training conditions. To confirm our intuition, we quantitatively evaluate the benefit of the

proposed approach in two weak conditions: a few-shot setting on NUAA-SIRST dataset, and knowledge transfer from NUAA-SIRST to IRSTD-850.

Few-shot training - For this purpose, we trained the networks on 25, 35 and 45 images. We did not consider 15-shot setting as the baseline does not converge under this setting. For each few-shot setting, we train the detectors on three distinct folds, with no overlap between them. The results obtained on the test set defined in Section 5.2 are averaged over these three folds and the computed means are given in Table 5.4. It can be seen that in such a frugal setting, our method performs significantly better than the baseline. Indeed, the baseline struggles to achieve decent performance in few-shot settings, reaching only 56.6% of F1 score in the 45-shot setting. Both $\text{NFA}_{\mathcal{N}}$ and $\text{NFA}_{\mathcal{U}_2}$ formulations outperform the baseline by a large margin. For example, $\text{NFA}_{\mathcal{U}_2}$ improves the baseline by 53.2% in the 25-shot setting, and by 35% in the 45-shot setting. The performance achieved by $\text{NFA}_{\mathcal{N}}$ is even more impressive: it already achieves a F1 score of 93.6% when trained with only 25 images. The F1 score decreases by only 4% when dividing by more than 10 the number of training samples. We thus conclude that adding a NFA head to the baseline significantly improves its robustness towards frugal setting. As explained for the segmentation in Section 4.2.3, such performance in few-shot setting is achieved thanks to the *a priori* on the *unexpectedness* of the small targets in comparison with the background induced by the *a contrario* test.

Knowledge transfer - We propose to evaluate the generalisation ability of weights trained on NUAA-SIRST to a more difficult dataset, namely IRSTD-850. Table 5.5 provides the results of the knowledge transfer without any fine-tuning step. It is clear that including a NFA criterion, especially the pixel-level one $\text{NFA}_{\mathcal{N}}$, brings robustness to the network. Indeed, YOLOv7-tiny+$\text{NFA}_{\mathcal{N}}$ achieves a F1 score that is about 5% higher than the baseline YOLOv7-tiny. The object-level NFA head also brings robustness, though the margin is smaller. However, the SOTA segmentation networks still perform significantly better for transferring the knowledge from NUAA-SIRST to IRSTD-850.

These experiments suggest a good robustness and generalisation capacity of the NN that includes our pixel-level $\text{NFA}_{\mathcal{N}}$ criterion in the detection head. This is particularly interesting when dealing with data that slightly differs from the distribution of the training data: changing brightness, background, noise, etc.

5.2.3 Small object friendly YOLO baselines

In the IRSTD literature, very few methods rely on YOLO-type networks, even though several baselines have been adapted for small object detection. These

baselines aim to address two major shortcomings of YOLO pipelines:

- **The generic YOLO architectures are not well-suited for small objects** – Since detection is performed on low-resolution feature maps, target information gets lost, negatively impacting recall. One potential solution is to perform detection on higher-resolution feature maps, as proposed in YOLO-fine [95].

- **IoU-based metrics, which are used both to evaluate models and train them, are not well-adapted for small objects** – This is mainly because IoU is highly sensitive to slight deviations for small objects, and when the IoU is zero, the cost function provides no indication of the distance or error between the prediction and ground truth. An increasingly popular solution in the literature involves modelling bounding boxes as 2D Gaussians and estimating the distances between boxes by directly calculating the distance between their distributions [96, 97].

Note that these methods have been minimally explored for IRSTD [68]. Although we have demonstrated the effectiveness of NFA on generic YOLO networks, it is reasonable to question whether our module also improves YOLO baselines that are better suited for small object detection. To achieve this, we will consider two variants of YOLO adapted for small object detection:

1. **YOLOv7-tiny-1scale** – We remove the two low-level detection scales and retain only the high-level detection branch.

2. **YOLOv7-tiny + NWD** – As in [96], we introduce a Gaussian prior by modelling the bounding boxes as 2D Gaussian distributions and use the normalised Wasserstein distance (NWD) to evaluate the discrepancy between the predicted boxes and the ground truth.

Table 5.6 provides the obtained results. We can observe that small object-adapted baselines YOLOv7-tiny-1scale and YOLOv7-tiny + NWD significantly improve performance, with an increase of over 5% in F1 score compared to YOLOv7-tiny. The literature on small object detection thus appears highly beneficial for the detection of small targets. Even more impressive, by adding our NFA module to these improved baselines, we achieve even better performance, particularly on the IRSTD-850 dataset. Finally, by combining all these approaches, YOLOv7-tiny-1 scale+NWD+$\text{NFA}_{\mathcal{N}}$ surpasses the SOTA segmentation method DNANet on IRSTD-850 dataset by more than 1% in F1 score and almost 3% in AP. This is a first for a detection network, highlighting the importance of network design and the contribution of the NFA module, even on an already strong baseline.

Backbone init.	SIRST		IRSTD-850	
	F1	**AP**	**F1**	**AP**
SOTA IRSTD baselines				
DNANet	96.9	98.1	<u>91.4</u>	92.4
YOLO baselines				
YOLOv7-tiny	96.5	97.8	84.0	90.1
+NFA$_{\mathcal{N}}$	97.6$^{(+0.7)}$	98.3$^{(+0.2)}$	90.1$^{(-1.3)}$	94.1$^{(+1.7)}$
YOLO 1 scale				
YOLOv7-tiny-1scale	96.6	97.9	90.5	93.6
+NFA$_{\mathcal{N}}$	**98.1**$^{(+0.5)}$	<u>98.5</u>$^{(+0.1)}$	91.3$^{(-0.1)}$	**95.4**$^{(+3.0)}$
YOLO + Gaussian prior (NWD)				
YOLOv7-tiny + NWD	97.4	98.0	89.9	94.2
+NFA$_{\mathcal{N}}$	<u>97.9</u>$^{(+0.8)}$	**99.0**$^{(+0.6)}$	90.8$^{(-0.6)}$	95.2$^{(+2.8)}$
Best ++				
YOLOv7-tiny-1 scale+NWD+NFA$_{\mathcal{N}}$	97.5$^{(+0.4)}$	98.4$^{(+0.0)}$	**92.5**$^{(+1.1)}$	<u>95.3</u>$^{(+2.9)}$

Table 5.6: Object-level metrics (F1, AP) achieved by methods adapted for small object detection on SIRST and IRSTD-850 datasets. Best results are in bold and second best results are underlined.

5.2.4 Generalisation to vehicle detection

Method	F1	AP	Prec.	Rec.
YOLOv7-tiny	68.4$^{\pm 2.3}$	72.5$^{\pm 3.0}$	74.6	63.2
+ **NFA**$_{\mathcal{N}}$	76.3$^{\pm 0.4}$	81.9$^{\pm 0.5}$	75.9	76.9
+ **NFA**$_{\mathcal{U}_2}$	71.0$^{\pm 0.9}$	75.9$^{\pm 0.7}$	74.4	68.1

Table 5.7: Object-level metrics (F1, AP, Prec., Rec.) achieved by the compared methods on VEDAI. Best results are in bold and second best results are underlined.

The NFA heads were specifically designed for very small objects that do not present a specific structure, and they have proved in previous paragraphs to bring robustness and even outperform the baseline in this case. We want to challenge our methods and apply them to a different scenario where the targets are larger (but still occupy a small percentage of our image) and present a geometric structure that is easily identifiable to the naked eye. To do this, we consider the publicly available VEDAI dataset [10], which is used for benchmarking vehicle detection in aerial images. It is composed of 1200 RGB images and their associated infrared

images, both of size 512×512. We split the VEDAI dataset into training, validation and test sets using a ratio of 60 : 20 : 20.

Table 5.7 presents the results obtained on the VEDAI dataset, and we can see that both the pixel-level and object-level NFA improve the baseline YOLOv7-tiny. Indeed, YOLOv7-tiny+NFA$_{\mathcal{U}_2}$ increases the F1 score and the AP by about 3% while also reducing the standard deviation, meaning that the network is more robust to the weight initialisation. The pixel-level NFA$_{\mathcal{N}}$ further improves the results, by increasing both criteria by at least 8%.

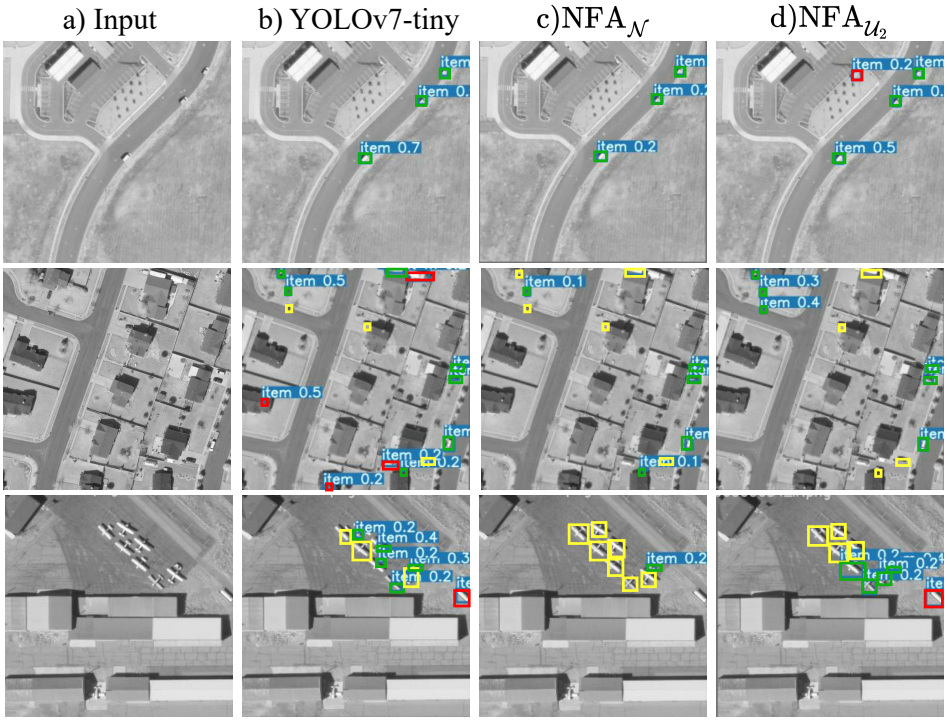
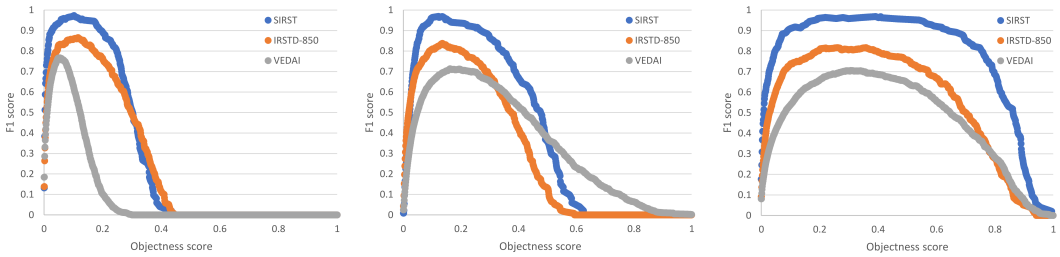


Figure 5.8: Qualitative results obtained with YOLOv7-tiny, YOLOv7-tiny+NFA$_{\mathcal{N}}$ and YOLOv7-tiny + NFA$_{\mathcal{U}_2}$ on the VEDAI dataset. Good detections, missed detections and false positives are framed in green, yellow and red, respectively.

Figure 5.8 shows some predictions on the VEDAI dataset. We can notice that the baseline leads to more detections, but also to many more false alarms than the methods including the NFA criteria. This is especially true for the images that tackle multiple object detection, as evidenced by the second line. From Figure 5.8 we can also notice that YOLOv7-tiny+NFA$_{\mathcal{N}}$ struggles to detect multiple large objects. For example, if we look at the third line, we can see that YOLOv7-tiny and YOLOv7-tiny+NFA$_{\mathcal{U}_2}$ can detect several planes, while YOLOv7-tiny+NFA$_{\mathcal{N}}$



(a) $\mathbf{NFA}_{\mathcal{N}}$ (b) $\mathbf{NFA}_{\mathcal{U}_2}$ (c) Baseline

Figure 5.9: F1 score dynamics depending on the objectness scores, averaged over SIRST, IRSTD-850 and VEDAI datasets, for a) $\text{NFA}_{\mathcal{N}}$, b) $\text{NFA}_{\mathcal{U}_2}$, and c) the baseline.

only detects the small vehicle next to the planes, but not the planes themselves. The fact that the $\text{NFA}_{\mathcal{N}}$ detection head leads to particularly poor performance for large object detection is consistent with its formulation: the *a contrario* paradigm improves the detection of *unexpected* events and thus assumes that the objects to be detected are anomalies. Since large objects (or multiple small objects) can represent a significant portion of the image, they are no longer considered as *unexpected* events by our $\text{NFA}_{\mathcal{N}}$ detection head.

The $\text{NFA}_{\mathcal{U}_2}$ formulation is more robust than $\text{NFA}_{\mathcal{N}}$ version for large object detection since at equal densities, a larger object is more significant than a smaller one. This phenomenon can be seen on Figure 5.9, which represents the F1 score dynamics depending on the objectness score for each method for several datasets. We can notice that for the $\text{NFA}_{\mathcal{N}}$ detection head, the larger the objects are in the datasets (e.g., the VEDAI dataset), the lower the objectness scores will be. We observe the opposite trend for the $\text{NFA}_{\mathcal{U}_2}$ formulation, where higher objectness scores can be achieved on the VEDAI dataset. The baseline does not seem to be sensitive to this criterion. This suggests that when dealing with medium-sized objects only (the VEDAI dataset still has a high proportion of very small cars), the $\text{NFA}_{\mathcal{U}_2}$ formulation may be more efficient.

5.2.5 Ablation study

Table 5.8 presents the ablation study made on some components of our different NFA modules. More specifically, we study the benefits of using a MC formulation and also of using channel attention (ECA) for weighting the maps obtained at different scales. This study is performed for the three versions of our NFA modules, namely using the $\text{NFA}_{\mathcal{N}}$, the $\text{NFA}_{\mathcal{U}_1}$ and the $\text{NFA}_{\mathcal{U}_2}$ formulations. We also provide the results achieved by the $\text{NFA}_{\mathcal{U}_1}$ head with a single channel as input (NFA SC column), which confirms the benefit of giving a multi-channel feature map as an

NFA	NFA SC	MC	ECA	F1	AP
$\mathbf{NFA}_{\mathcal{N}}$		✓		$97.6^{\pm 0.8}$	$98.3^{\pm 0.1}$
$\mathbf{NFA}_{\mathcal{N}}$		✓	✓	$97.6^{\pm 0.3}$	$98.3^{\pm 0.1}$
$\mathbf{NFA}_{\mathcal{U}_1}$	✓			$96.0^{\pm 0.8}$	$97.5^{\pm 0.5}$
$\mathbf{NFA}_{\mathcal{U}_1}$		✓		$96.4^{\pm 0.6}$	$98.0^{\pm 0.2}$
$\mathbf{NFA}_{\mathcal{U}_1}$	✓		✓	$96.3^{\pm 1.1}$	$98.0^{\pm 0.2}$
$\mathbf{NFA}_{\mathcal{U}_1}$		✓	✓	$96.8^{\pm 1.6}$	$98.1^{\pm 0.3}$
$\mathbf{NFA}_{\mathcal{U}_2}$				$96.5^{\pm 1.4}$	$98.1^{\pm 0.2}$
$\mathbf{NFA}_{\mathcal{U}_2}$		✓	✓	$97.0^{\pm 0.3}$	$97.8^{\pm 0.3}$

Table 5.8: Ablation study performed on NUAA-SIRST. We evaluate (using object-level metrics) the different formulations of NFA ($\text{NFA}_{\mathcal{N}}$, $\text{NFA}_{\mathcal{U}_1}$, and $\text{NFA}_{\mathcal{U}_2}$) and compared the benefits of MC and using channel attention (ECA) in our NFA modules. We also provide the results of $\text{NFA}_{\mathcal{U}_1}$ head when a single-channel feature map (NFA SC) is provided as an input of this module.

input for our $\text{NFA}_{\mathcal{U}_1}$ head. We provide results averaged over three runs on NUAA-SIRST dataset.

The $\text{NFA}_{\mathcal{N}}$ formulation is already multi-channel, so we are only analysing the contribution of attention per channel. As a reminder, by choosing a constant number of tests η_{test} as discussed in Section 3.3 of Chapter 3, each spatial scale has the same impact on the decision: the NN detects both small objects (fine resolution scale) and larger objects (low resolution scale), without any preference. However, depending on the image or application, we might want to favour the detection of objects of larger or smaller size. This is why we have proposed the use of a channel-based attention module to give each spatial scale a weighting that favours (or not) the detection of objects of a given size. From Table 5.8 we can notice that using ECA layer does not improve the performance for the $\text{NFA}_{\mathcal{N}}$, however it seems to bring robustness in the results, since we observe a small decrease in the standard deviation.

For the $\text{NFA}_{\mathcal{U}_1}$ version, ECA layer significantly improves the performance. Moreover, introducing a multi-channel version of this NFA formulation also improves the performance, with an F1 score increased by 0.4%. The combination of both MC and ECA leads to best performance for the $\text{NFA}_{\mathcal{U}_1}$ formulation.

Since we have empirically demonstrated the contribution of MC and ECA independently on the $\text{NFA}_{\mathcal{U}_1}$ formulation, we present only the results of MC and ECA combined for the $\text{NFA}_{\mathcal{U}_2}$ version. The last line of Table 5.8 confirms the contribution of these two elements to the robustness of the training. The F1 score is also improved by 0.5%.

Overall, multi-channel formulations and the use of channel-attention to auto-

matically weight the different spatial scales improve and stabilise the performance of each NFA version. These conclusions are consistent with those observed in the Chapter 4.

5.3 Conclusion

In this chapter, we proposed to integrate different *a contrario* criteria into the detection head of a YOLO network, in order to re-estimate the objectness scores provided by YOLO. Specifically, we have designed three detection heads based on three different NFA formulations: the $\text{NFA}_{\mathcal{N}}$, whose naive model is a normal distribution, and two versions based on a uniform naive model, namely $\text{NFA}_{\mathcal{U}_1}$ and $\text{NFA}_{\mathcal{U}_2}$. The latter two versions are used to compute a NFA at the object level (i.e., on the predicted bounding box) and allow for better modelling of the relationships between the predicted bounding boxes and their associated objectness scores. Evaluations on SIRST, IRSTD-850 and VEDAI datasets have confirmed the benefits of the *a contrario* paradigm for detecting small objects using a YOLO backbone. This is even more true under challenging conditions, such as in a frugal context or when transferring knowledge from SIRST to IRSTD-850. Although the object-level versions of the NFA, in particular $\text{NFA}_{\mathcal{U}_2}$, improve the YOLO baseline in most cases, it is the $\text{NFA}_{\mathcal{N}}$ version, introduced in Chapter 4, that is more robust and performs best on all benchmarks. Furthermore, YOLOv7-tiny + $\text{NFA}_{\mathcal{N}}$ bridges the performance gap with SOTA segmentation methods for detecting small targets. Even more impressive, adding our NFA module on top of a YOLO baseline adapted for small object detection leads to new SOTA results. We will retain this NFA formulation as the most efficient for IRSTD, and in the following any reference to a NFA will refer directly to $\text{NFA}_{\mathcal{N}}$.

This chapter concludes Part I, in which the benefits of the *a contrario* paradigm for small target detection were demonstrated through a large number of experiments. By integrating the NFA test, in particular $\text{NFA}_{\mathcal{N}}$, into the training loop of a segmentation network and then into a detection network, we have not only improved the respective baselines, but also obtained results that are very competitive with the SOTA networks for infrared small target detection. The following 5.9 and 5.10 tables provide an overview of the performance achieved by the methods we developed in Part I on two IRSTD datasets, namely SIRST and IRSTD-850. Table 5.9 shows that both DNIM + $\text{NFA}_{\mathcal{N}}$ and YOLOv7-tiny + $\text{NFA}_{\mathcal{N}}$ outperform the SOTA network DNANet on the SIRST dataset. Regarding the results obtained on the IRSTD-850 dataset, DNANet and DNIM + $\text{NFA}_{\mathcal{N}}$ perform on par, while generic YOLO-based methods yield poorer results on this challenging dataset. Nevertheless, integrating the *a contrario* criterion into YOLOv7-tiny allows us to improve the baseline by a large margin and to reduce the performance

Method	NUAA-SIRST		IRSTD-850	
	F1	AP	F1	AP
DNANet	<u>96.9</u>	<u>98.1</u>	<u>91.4</u>	92.4
DNIM + $NFA_{\mathcal{N}}$	**97.6**$^{(+0.7)}$	**98.4**$^{(+0.3)}$	91.3$^{(-0.1)}$	**94.2**$^{(+1.8)}$
YOLOv7-tiny	96.5	97.8	84.0	90.1
YOLOv7-tiny+$NFA_{\mathcal{N}}$	97.6$^{(+0.7)}$	98.3$^{(+0.2)}$	90.1$^{(-1.3)}$	94.1$^{(+1.7)}$
YOLOv7-tiny-1 scale+NWD+$NFA_{\mathcal{N}}$	97.5$^{(+0.6)}$	98.4$^{(+0.3)}$	**92.5**$^{(+1.1)}$	<u>95.3</u>$^{(+2.9)}$

Table 5.9: Results obtained on SIRST and IRSTD-850 datasets. The best results are in bold, the second best results are underlined and the improvement over the baselines is indicated in the superscript.

gap with SOTA segmentation methods. Furthermore, adding our NFA module on top of a specific and strong YOLO baseline (namely YOLOv7-tiny-1 scale+NWD) leads to a very efficient method that outperforms the SOTA segmentation baseline for IRSTD. Last but not least, YOLOv7-tiny + $NFA_{\mathcal{N}}$ outperforms both baselines and DNIM + $NFA_{\mathcal{N}}$ in a frugal setting, as shown by Table 5.10. This confirms the relevance of using an *a contrario* criterion to improve the performance of baselines under very difficult training conditions, such as a frugal context.

Method	25-shots	
	F1	**AP**
DNIM	87.0	82.6
DNIM + $NFA_{\mathcal{N}}$	<u>90.9</u>$^{(+3.9)}$	<u>93.1</u>$^{(+10.4)}$
YOLOv7-tiny	21.8	15.0
YOLOv7-tiny + $NFA_{\mathcal{N}}$	**93.6**$^{(+71.8)}$	**95.0**$^{(+80.0)}$

Table 5.10: Results achieved in a 25-shot setting on SIRST dataset. Best results are in bold, second best results underlined and the improvement over the baselines is indicated in the superscript.

In the light of the results and conclusions drawn in this first part, we can identify two main short-term perspectives:

- **Improving small target detection without impairing the detection of larger or more numerous objects** – Indeed, we have seen that $NFA_{\mathcal{N}}$ detection and segmentation heads struggle with the detection of larger objects. More specifically, when the objects to be detected occupy a fairly large portion of the image, the *significance* scores output by our $NFA_{\mathcal{N}}$ module drop sharply, which can lead to many missed detections. Several strategies

can be explored to overcome these limitations. For example, the background statistic can be estimated more robustly from a larger number of samples, rather than from a single image. Another solution is to rely on ensemble methods to merge the predictions of two (or more) detectors: one specialised in detecting large objects (e.g., the common object detection baselines), the other specialised in detecting small objects (e.g., using a NFA detection head).

- **Adapting the YOLO backbone for small object detection** – As observed on IRSTD-850 dataset, generic YOLO backbone is not as efficient as SOTA segmentation baselines, even when using the $\text{NFA}_{\mathcal{N}}$ detection head. The use of strong YOLO baselines that are specific to small object detection have proven to significantly improve the results, leading to SOTA results when using our NFA detection head. This shows the importance of the encoder design, although the use of our NFA module with a generic YOLO backbone already leads to good performance. Another idea could be to boost the network training by using appropriate pre-trained weights for the encoder. This is what we propose to do in the following section with the use of weights pre-trained in a self-supervised manner.

Part II

Self-supervised learning and small object detection

Chapter 6

A survey on self-supervised learning for image representation learning

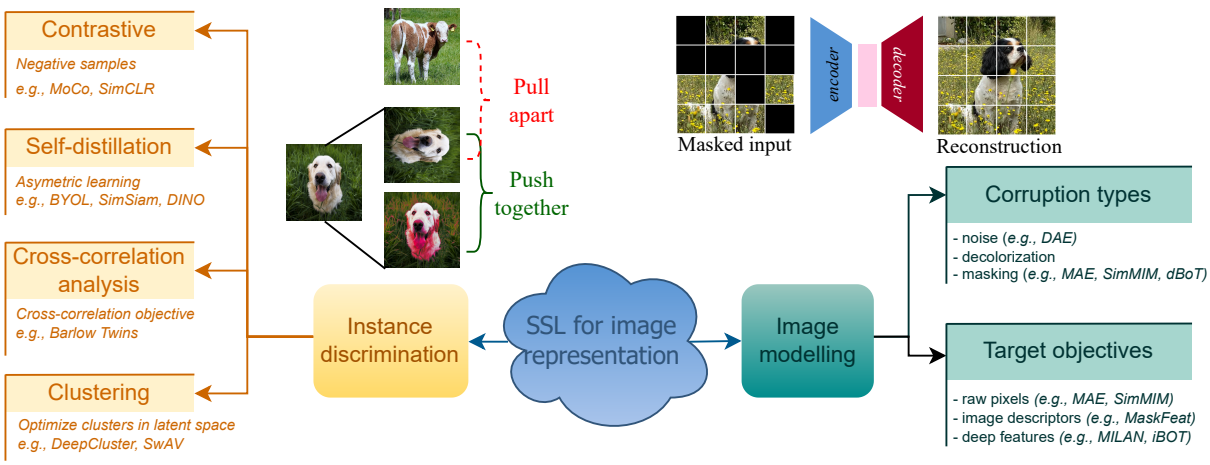


Figure 6.1: SSL methods for image representation learning.

This chapter is the first in Part II, where we investigate the usefulness and the potential benefits of SSL for small target detection. Specifically, SSL consists in an unsupervised training of deep learning networks (often only the encoder) using a well-designed pretext task. The aim of this pre-training task is to help the network learning features or invariances that are relevant for the final task (also called the downstream task). One of the motivations comes from the fact that SSL methods have been shown in the literature to improve SOTA performance for many use cases. More specifically, SSL allows the network to learn general features from large unlabelled datasets which, when transferred to a final task, will improve performance despite difficult fine-tuning conditions (e.g., little annotated

data or few computational resources). However, the use cases considered in the literature mainly concern classification or the detection of objects of medium to large size. The following question arises: do these conclusions transpose to the detection of small targets too? And under what conditions? In this first chapter of Part II, we propose a survey of the different SSL methods that exist for image representation learning in general. This survey is not specific to object detection (and in fact, these methods are often designed for classification), but it provides an introduction to most of the paradigms and issues that arise in SSL. Several recent surveys on SSL for image representation learning have been proposed in the literature [98, 99, 100, 101]. We present the SSL from another point of view, with a particular focus on masked Image Modelling (IM) methods compared to existing surveys. The general taxonomy we propose in Figure 6.1 is closer to the one proposed in [100, 101]. Indeed, we claim that the SSL pretext tasks can be divided into two main categories: instance discrimination methods, and image modelling. We will present these two approaches in detail in the following sections.

Then, in Chapter 7, we extend this survey (and thus complete what is proposed in the literature) by proposing a survey specific to SSL methods for object detection. We also propose to benchmark different SSL methods on several datasets, with a focus on small object detection.

6.1 Instance discrimination

Instance discrimination methods aim at modelling the decision borders between sub-sets of data represented in the latent space. Fundamental methods considered images as instances, and perform inter-image discrimination, as illustrated on Figure 6.2. Concretely, the optimisation objective consists in minimising, in the latent space, the distance between features of instances that share similar semantic properties. Given two transformed images, if these images are augmented views (called positive samples or pairs) of the same anchor image, we consider that they share the same semantic property and thus we force their features to be represented similarly in the latent space. Teaching a network to identify whether two images come from the same anchor image forces the encoder to learn general and representative features of a given image while being invariant to the augmentations used to create the augmented views.

The choice of the augmentation transforms is therefore crucial: it must not be too simple (i.e., close to the identity function), without completely distorting the two images, in order to preserve their semantic similarity. Common data-augmentations include crop and resize, random rotation, color jittering, gaussian noise and gaussian blur. Some of them are represented on Figure 6.3. More sophisticated data-augmentations include CutOut [102], CutMix [103] and MixUp [104].

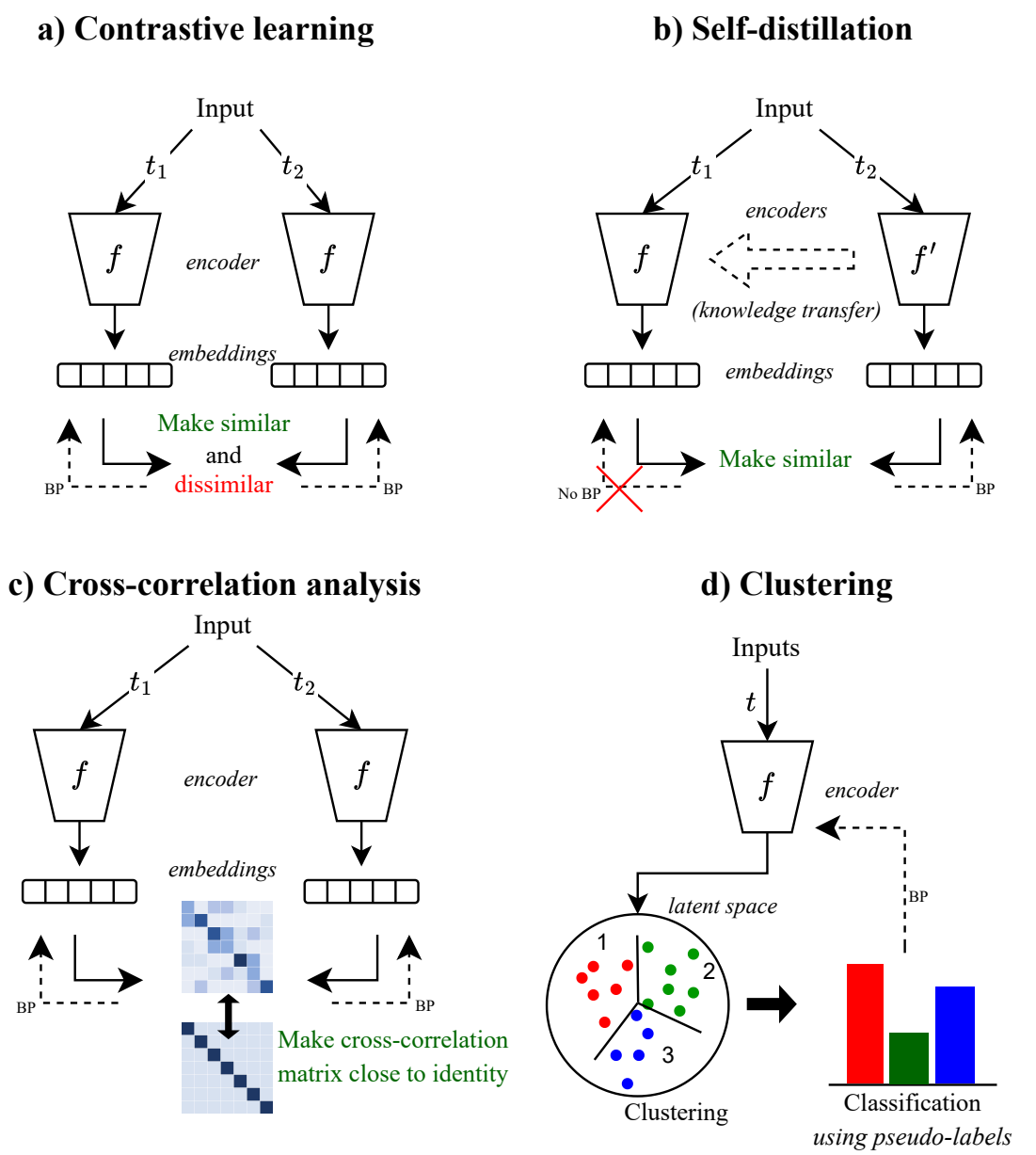
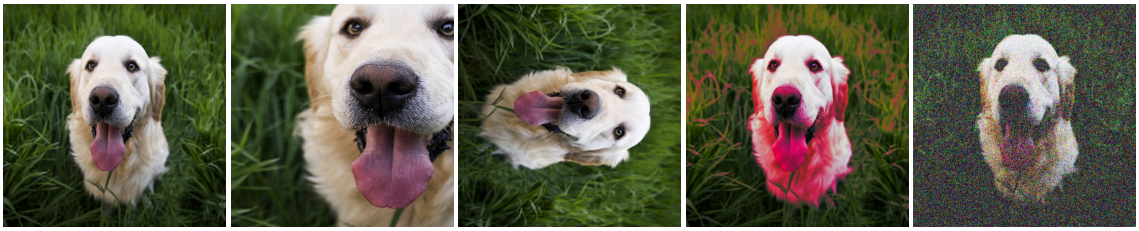


Figure 6.2: Instance discrimination methods. "BP" stands for backpropagation, and t, t_1 and t_2 are different data-augmentation transforms.



a) Original image b) Crop + resize c) Rotation d) Color jittering e) Gaussian noise

Figure 6.3: Common data-augmentations for SSL.

One criterion for choosing good augmentations can be based on the perceptual invariances that are needed to solve the final task. The types of invariance include invariance to color, illumination, occlusion, viewpoint, etc., and they are not all useful for a given final task. For example, if we want to discriminate red apples from green ones, colour jittering should be used carefully to create augmented views of the anchor image as it is a discriminant feature.

One challenge when dealing with such methods is to avoid the trivial solution where all images are represented by a constant representation (in such case, the network is said to have *collapsed*). Although the trivial solution satisfies the training objective (i.e., the features extracted from augmented views are indeed similar since they are equal), no general representative information about the images has been learned by the neural network. Four solutions have been proposed to prevent from such phenomena: i) considering negative pairs with contrastive learning, ii) using asymmetric siamese networks with self-distillation, iii) analysing the correlation between latent variables, and iv) introducing constrained clustering. Figure 6.2 presents their respective architecture, and we present the the details of each method in the next sections.

6.1.1 Contrastive learning

Contrastive learning is among the most popular self-supervised learning methods for instance discrimination. It comes from deep-metric learning, whose goal is to bring similar data points close together and dissimilar data points far apart in the latent space. As illustrated on Figure 6.2a, it relies on the use of a symmetric siamese network. Like for any instance discrimination method, the embeddings extracted from the positive pairs of samples are made similar through the objective loss. Contrastive learning also introduces some negative samples that are explicitly made dissimilar to the positive samples. This prevents the network from collapsing: indeed, the representation of any negative sample is forced to be different from the positive sample representations, thus preventing the trivial solution. In general, the positive samples are augmented views of a given image while the negative

images are randomly/arbitrarily sampled from the rest of the dataset, as shown on the left illustration of Figure 6.1.

Common architectures – MoCo [105] and SimCLR [106] are two emblematic contrastive self-supervised learning methods which form the basis of several SOTA SSL methods. SimCLR consists of a symmetric siamese network. Image representations of two augmented batches of images are computed by an encoder (encoders share the same weights between the two branches) and a projector, and then are fed to the contrastive loss. Although being very simple and effective, SimCLR requires very large batches as inputs (often more than several thousands) to achieve good performance. The use of a memory bank (dictionary of features of all samples) as in PIRL [107] is a good solution to accumulate negative samples, however image representations are often outdated. This issue has been solved by MoCo, which encodes and updates on-the-fly a queue (dynamic dictionary) of negative samples (not all samples) using a momentum encoder. This ensures representation consistency across all negative samples as the representations are updated smoothly at each step.

Other contrastive frameworks mainly rely on MoCo or SimCLR, and they either add some regularisation terms or consider specific data-augmentations. For example, PIRL uses jigsaw puzzle as a positive view. The network is thus forced to be invariant to patch permutation, and the authors argue that such invariance allows them to maintain semantic information in the representation. RELIC [108] adds a regularisation term that ensures the predictions to be invariant across different augmentations. RELICv2 [109] introduces some background invariance by masking background elements (foreground elements are estimated using salience maps). Some frameworks are also specifically designed for anomaly or out-of-the-distribution detection. CSI [110] considers distribution shifting transformations like CutOut, rotations or patch permutation as negative augmentations. Spot-the-Difference [111] adds a regularisation term to MoCo or SimCLR that maximises the distance between a sample and an augmented version with an anomaly incrustation.

Data-augmentation matters – Like with any instance discrimination method, a major challenge lies in the choice of the augmentations for creating the positive and negative pairs. [106] shows that, for classification on ImageNet, composing random cropping and random colour distortion significantly improves the quality of the representation. It also prevents the network from exploiting some shortcuts, and thus encourages the learning of generalisable features. However, it is worth noting that the definition of a positive or negative sample is dependent of the nature of the final task: for example, [110] shows that considering shifting

transformation (e.g., CutOut or rotations) to create negative samples while using very weak augmentations for positive pairs is beneficial for out-of-the-distribution task. In other cases, these transformations can be used to create positive pairs, in particular when we want to introduce invariances to rotations or occlusions. The definition of a negative or positive pair is therefore highly dependent on the invariances we want to teach the network. It is also possible to select appropriate augmentation policies by using an automatic augmentation selection algorithm such as AutoAugment [112] or SelfAugment [113].

Contrastive loss and SSL – The contrastive loss [114] was first introduced in the context of deep metric learning. Concretely, it takes a pair of input embeddings z_i and z_j and predicts whether or not they belong to the same (pseudo-)class. It can be written as follows:

$$L_{\text{contrastive}}(z_i, z_j) = \mathbb{1}_{\{y_i=y_j\}}||z_i - z_j||_2^2 + \mathbb{1}_{\{y_i\neq y_j\}} \max(0, m - ||z_i - z_j||_2)^2,$$

where y_i and y_j are the labels of z_i and z_j respectively, and $m > 0$ is a margin parameter. A similar loss, called the Triplet loss [115], was also introduced. Unlike the contrastive loss, it compares three samples at the same time: a query, a positive sample and a negative sample. Although hard negative mining can help, one problem with such losses is their slow convergence.

In order to take into account multiple negative samples, [116] extends the triplet loss to a N-pair loss, which boils down into a multi-class classification loss. Adding a temperature scaling parameter τ that controls the penalty applied to hard negative samples in the N-pair loss leads to the InfoNCE loss [117]. The latter is widely used in SSL frameworks, and can be formulated as follows:

$$L_{\text{InfoNCE}}(q) = -\log \frac{\exp(q \cdot k_+/\tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i/\tau)}, \tag{6.1}$$

where q is the query embedding, $\{k_0, k_1, ..., k_K\}$ a given batch of samples containing K negative samples with respect to q and one positive sample, also denoted k_+.

Computational resources – A drawback of contrastive methods is that they are computationally expensive. They require an important number of negative samples, leading to very large training batches. To understand the reasons behind these requirements, we need to go back to the fundamental challenges of deep metric learning.

In common deep metric learning methods, the negative samples are meticulously chosen. Indeed, in order to help the network to extract more representative

information, the hard negative sampling has been thoroughly studied in deep metric learning. Good hard negatives are negative samples that are considered as being difficult to discriminate from the positive samples. However, depending on the dataset and other biases, using only hard negatives as negative samples can make it difficult for the network to converge. Some strategies, such as semi-hard negative sampling [118], were thus introduced. Contrastive learning circumvents this issue by relying on large batches of uniformly sampled negatives, which implicitly ensures that at least some hard negatives are computed in the loss. Nonetheless, such strategy requires large datasets, and also important computational resources (GPU memory). As explained earlier, some methods alleviate this issue by using a memory bank (e.g., PIRL), or momentum encoder (e.g., MoCo). MoCHi proposes to reduce the number of training epochs by relying on synthetic hard negative mining. More specifically, they mix hard negative features together in order to create synthetic hard negatives. Such a strategy also increases the downstream task performance. However, low-resource SSL pre-training remains a major challenge for contrastive learning.

Uniformity-tolerance dilemma – Contrastive learning brings several advantages: the architecture is quite simple, and allows for the use of any type of network as an encoder (e.g., ConvNet or ViT). Moreover, the choice of the data-augmentations for generating positive pairs is intuitive and can be designed specifically for different downstream tasks. However, its learning objective induces some noticeable drawbacks. [119] points out a uniformity-tolerance dilemma in the formulation of the contrastive learning objective, which may impair the fine-tuning performance. Indeed, the training objective of contrastive learning leads to a uniform distribution of the embeddings in the latent space in order to learn separable features. However, as positive pairs are built upon data-augmentations of one anchor sample only and negative pairs are randomly sampled, giving a high penalty to hard negatives implies that the network will ignore the underlying semantic consistency between hard negatives and positive samples. Here is an example: let us say that we have a Golden Retriever (dog) as our anchor sample, and a Labrador (dog) and an apple as negative samples. In the latent space, the Labrador and the apple will both be pushed far away from the augmented views of the Golden Retriever, though we expect the Labrador to have a feature representation similar to the one of the Golden Retriever since they both belong to the same class (dog). With such training objective, the network may learn less representative features. [119] argues that a careful choice of the temperature τ in the InfoNCE loss can relax such dilemma: using a higher τ reduces the penalty applied to hard negatives (such as the Golden Retriever). Indeed, as described in the Figure 2 of [119], a small temperature τ leads to a more uniform distribution among the

embeddings in the latent space, and thus induces less tolerance towards hard negatives. [120] further analyses this phenomenon and proposes to introduce a dual temperature that penalises differently hard negatives. Besides, this strategy allows us to remove the dictionary used in MoCo (method coined as SimMoCo) and its momentum encoder (method coined as SimCo), without impairing the performance. Another efficient solution to the uniformity-tolerance dilemma would be to introduce multi-instance positives. This can be easily achieved when using video sequences, which allows one to have different viewpoints of a same object [121] and thus induces viewpoint invariance. However, videos are not always available for the pretext task pre-training. To circumvent this issue, NNCLR [122] proposes to create multi-instance positive pairs by picking a positive view of an anchor in a memory bank. The positive sample is chosen as being the sample with the closest features to the anchor image using the l_2 distance. Negatives are stored in the mini-batch as in SimCLR. Finally, [123] introduced a debiased contrastive objective, which limits the effects of the sampling bias (i.e., false negative samples) on the SSL training in an unsupervised way.

6.1.2 Self-distillation

Another way to circumvent the uniformity-tolerance dilemma is to remove the negative pairs and to consider positive pairs only, although some tricks need to be introduced in order to prevent the network from collapsing.

One of the first strategies to train such networks is introduced in [124] with BYOL. As illustrated in Figure 6.2b, it consists in distilling the knowledge from a teacher branch to a student one (i.e., training the student network to predict the representations learned by the teacher). To prevent the network from collapsing, the teacher's weights are not shared with the student branch (i.e., there is no backpropagation): instead, the teacher's weights are progressively updated through an Exponential Moving Average (EMA) of the student's weights. The training asymmetry induced by such update is the key to avoid trivial solution. More specifically, SimSiam [125] shows that the stop-gradient operation (i.e., no backpropagation in the teacher branch) coupled with a predictor in the student branch is enough to prevent from collapsing. However, the use of EMA to update teacher's weights still leads to better performance.

SOTA frameworks based on self-distillation are built upon BYOL and SimSiam. For example, DINO [126] improves BYOL by smoothly discretising the representations. The authors also show that multi-crop augmentations are essential for improving the fine-tuning performance, and that ViT backbones trained within an SSL framework implicitly learn local information about the scene such as object boundaries (it can be seen by looking at attention maps). iBOT [127] and DINOv2 [128] further improve DINO by integrating masked image modelling

task (described in Section 6.2). Note that for self-distillation frameworks, the use of a contrastive loss is not adapted since we do not have negative samples; losses like the mean-squared error (MSE), the cosine similarity or the cross-entropy are used instead.

6.1.3 Cross-correlation analysis

Methods described in this paragraph rely on cross-correlation analysis, and they neither require the use of negative pairs nor an asymmetric siamese network. Instead, they are based on a symmetric siamese network that maximises the mutual information (invariance term) of decorrelated latent representations of positive pairs, as illustrated by Figure 6.2c). To prevent from collapsing, such methods propose to integrate the following properties into the loss: i) decorrelate the pairs of positive embeddings (covariance term), and ii) ensure a non-zero variance across the batch (variance term). Barlow Twins [129] achieves this by computing the cross-correlation matrix of normalised positive embeddings (normalisation performed across the batch dimension), and by making it close to the identity matrix. W-MSE [130] ensures the variance and covariance properties by whitening (i.e., decorrelate data and force them to have a unit variance, which is equivalent to enforce spherical distribution) the batch before ensuring the invariance property (i.e., making the embeddings similar). VicReg [131] improves Barlow Twins by explicitly optimising the variance, invariance and covariance with separate terms in the loss. Although such methods consist in simple architectures, the complex nature of the introduced losses can lead to sub-optimal or difficult optimisations.

6.1.4 Clustering

Clustering data points in a latent space is the most straightforward method to perform instance discrimination, i.e. to model the decision boundaries between groups of images sharing similar features. Unlike common similarity based methods (except for NNCLR), clustering enables to perform multi-instance discrimination, i.e. images originating from different anchor points can be clustered together in the objective function. In a self-supervised learning framework, clustering can be used to estimate the clusters of data and assign pseudo-labels for each group, as shown in Figure 6.2d. One of the pioneering work to perform this is Deep Cluster [132]: after encoding a batch of images, a clustering assignment is performed using simple clustering methods such as K-means, which allows us to assign a pseudo-label for each image. Then, a classification loss is optimised. Several issues arise from this formulation: i) clusters derived from features extracted from randomly initialised weights seem to be *meaningless* (i.e., no prior about the semantic consistency is ensured, as it would be the case for similarity-based methods with the use of positive

pairs), ii) the number of clusters needs to be chosen beforehand, and iii) collapsed solutions where all data points collapse to a single cluster can occur. For the first issue, [132] argues that NN trained with DeepCluster can converge although starting from random initialisation since randomly initialised weights already provide a strong prior on the input signal. Indeed, some simple low-level features can be discriminated by random weights. Concerning the number of clusters, DeepCluster empirically show that a 10k clusters help in achieving best performance. For the third issue, authors of [132] avoid trivial solution by reassigning empty clusters (perturbed centroid of non-empty cluster becomes new centroid for empty cluster). They also balance the size of clusters by sampling images uniformly among the estimated pseudo-labels. SeLa [133] improves Deep Cluster by i) constraining the clusters to be equally-sized in order to tackle the third issue, and ii) interpreting the final objective as an optimal transport problem and solving it efficiently using the Sinkhorn-Knopp algorithm.

Some more methods based on offline clustering were proposed (such as SCAN [134]), however these methods find difficulties in scaling to huge amounts of data. Moreover, some hyperparameters such as the number of clusters need to be tuned carefully to achieve great performance. To circumvent these drawbacks, SwAV [135] has been proposed, which takes advantage of both clustering and similarity-based methods. To do so, the symmetric siamese architecture of SwAV takes as input two augmented batches of images (i.e., positive pairs), assign a prototype vector to each image and make them similar for positive pairs. So far, SwAV is considered as being one of the most stable clustering method, and as it combines the best from both clustering and similarity-based approach, it leads to SOTA results. A similar method called MSN [136] combines self-distillation, online clustering (with prototype vectors) and masked augmentations, which also leads to SOTA results.

6.2 Image modelling

Conversely to instance discrimination methods whose goal is to estimate decision borders between image representations, the objective of IM methods is to recover corrupted images. The underlying hypothesis is that if a network is able to guess or even reconstruct severely corrupted information, then it "understands" the semantics in the image. One strong invariance learned by such methods is occlusion invariance, which has been shown to benefit instance discrimination methods [137].

The pioneering methods that deal with image modelling firstly aimed at predicting the reconstruction parameters. For example, the pretext task proposed in Context Prediction [138] consists in predicting the relative position between two patches. To solve this task, the network needs to recognise the objects and how their parts are related to each other. The same requirements are needed for solv-

ing a jigsaw puzzle [139] (i.e., predicting the applied permutation). RotNet [140] pretext task consists in applying a random rotation (0, 90, 180 or 270 degrees) to an image and to predict it. This supposes that the network learns about the semantics and the "natural" relationships between the elements composing an image/a scene. For example, it should learn that, usually, the head of an animal is placed above the body, not below it.

The methods outlined above do not reconstruct the corrupted image, which means that they are less computationally expensive. However, their design is very dependent of the dataset biases and they are prone to shortcuts. For example, [139] showed that a common shortcut for solving jigsaw puzzles is the use of chromatic aberration. A careful design of the pretext task is crucial to ensure that relevant features are learned by the network. With the fast development and progress of deep learning frameworks, the task of corruption prediction became too simple to solve and these methods were quickly surpassed by methods that explicitly reconstruct the corrupted information. Their success is justified by the fact that a strong feature extractor should be robust to partial corruption of the data. Indeed, we are able to recognise a dog even if the picture is blurred or if a part of the dog is occluded. Moreover, by reconstructing the corrupted data, image modelling methods better exploit local features [141], compared with simple corruption perdition pretext task.

In the following, we present several "modern" IM methods. The key differences between IM methods lie in the corruption that is considered, as well as the reconstruction objectives. In the following paragraphs, we propose to classify the emblematic IM methods according to the chosen corruption and the reconstruction target objectives.

6.2.1 Corruption types

Figure 6.4: Common types of corruption for image modeling.

The main corruption types are represented on Figure 6.4. One of the first considered corruptions is adding noise to the input, which was then denoised using

a denoising auto-encoder (DAE) [142]. Concretely, DAE adds a gaussian noise at each scale of the auto-encoder and attempts to reconstruct the original features and image. [143] points out that such a strategy is very localised and learns low-level features, and thus no semantic knowledge seems to be required to solve such a task. [143] proposes to mask a part of an image with white pixels and to perform inpainting. Another corruption-based pretext task that has led to better performance than previous methods is image colourisation [144, 145]. The relationships between the objects and their colours are learned by the encoder, and colour thus becomes a discriminative feature for analysing image semantics. This means that this pretext task may not be adapted to 1) final tasks that need to be colour invariant, and 2) non-RGB images (e.g. infrared images). It is also necessary to ensure that the pre-training dataset is diverse enough, as this method is much more prone to colour biases in the dataset (e.g., only red apples are presented in the dataset, while green or yellow apples also exist).

Early image modelling methods yielded poor results compared to supervised training. The recent advent of ViT has turned the tables, and now Masked Image Modelling (MIM) methods are at the SOTA for image representation learning. MIM consists in masking a relatively high proportion of an image and reconstructing it (or its features). This brings occlusion invariance to the encoder, as well as locality inductive bias [141]. Well-known SSL SOTA pipelines such as BEiT [146], Masked Auto-Encoder (MAE) [147], iBOT [127] or I-JEPA [148] rely on this principle. However the fine-tuning performance is highly dependent on the masking strategy. We propose to group the masking strategies by answering the following questions:

Figure 6.5: Common masking strategies for masked image modelling.

- **What shape for the mask?**

  Authors from MAE [147] and SimMIM [149] evaluate different mask sampling strategies that were previously proposed in the literature, including random, square [143], block-wise [146] and grid masking strategies. The different masking strategies are represented on Figure 6.5. Both works conclude

that the simple random masking strategy is the most efficient, under the condition of considering a high masking ratio. Indeed, such a strategy preserves more hints about the object, especially when considering an object-centric dataset, as opposed to the square and blockwise strategies. Compared to the regular grid masking strategy, random masking brings more difficulties to the network since the object parts are unevenly occluded. Therefore, a semantic understanding of non-occluded patches is necessary to reconstruct some heavily occluded parts. A commonly chosen size for the masked patches is 32 when considering pre-training on images of size 224×224, which has shown to be efficient for many famous computer vision datasets (ImageNet [33], COCO [35], etc.). Note that such patch masking strategies (in terms of size and shape) may not be suitable for some real-world application and data, like in remote sensing or medical domains. [150] proposes masks with irregular shapes, which are beneficial for anomaly detection in remote sensing images because the authors simulate the spatial morphology of the anomalies. However, the effectiveness of this masking strategy compared to conventional strategies has yet to be proven in this type of application.

- **At which ratio?**

  Masking a high ratio of patches is also important to make the pretext task difficult enough for the network to extract meaningful features. MAE and SimMIM found that a ratio of 50% is optimal for random masking. SimMIM further proposes a metric called Average Distance (*AvgDist*) that evaluates the reconstruction difficulty of a given mask sampling strategy. It consists in computing the averaged Euclidean distance between masked pixels and the nearest visible ones. They conclude that masking strategies with an *AvgDist* metric between 10 and 20 have more chance to perform well for fine-tuning. Note that this study has been performed on object-centric datasets (ImageNet, iNaturalist-2018 [151]), as well as on visual scenes (COCO and ADE20K [152]).

- **Which values for the masked areas?**

  First transformer-based MIM methods propose to replace the masked patches by learnable embeddings [146, 149]. However, MAE showed that encoding masked patches leads to worse results: in addition to severely affecting the convergence time, it also brings a gap between pre-training and fine-tuning. Indeed, in the fine-tuning task, there is no such corrupted patches. Therefore, authors from MAE paper propose to encode unmasked patches only, and design a specific decoder that takes as input the masked patches as learnable embeddings. For ConvNets, basic masking values like white [143], zero or mean RGB [153] values are not optimal. Indeed, [154] explains that since

CNNs operate on regular grids with an overlapped sliding window, zero-outing masked pixels inevitably leads to i) data distribution shift, and ii) mask pattern vanishing issue. Therefore, the authors propose to rely on sparse convolutions, also called partial convolutions. [155] and [156] use a similar strategy. More specifically, [156] efficiently encodes two images as a single image by replacing the masked patches of the first image (image 1) with the unmasked ones of the second image (image 2). To adapt this strategy to ConvNets, they introduce unmixed convolutions, which consists in unmixing the image into image 1 and image 2, and then applying partial convolution.

Another strategy consists in replacing the masked patches by plausible patches. CIM [157] replaces the masking strategy by a more subtle corruption created using a generative network. Such masking strategy seems particularly appropriate for anomaly detection tasks, although the generation of subtle corruptions and their encrustation raise many questions.

- **Where?**

  Some papers observe that masking patches at random locations can impair the performance of the network [158]. Indeed, if the pre-training dataset contains small objects, they may be totally occluded. The objective of the network will therefore no longer be to reconstruct information but to hallucinate small objects, which poses a problem in terms of learning quality. To avoid this problem, several papers thus focus on optimising the masking strategy. [158] propose a conservative data transform to maintain clues about foreground objects. MST [159], AttMask [160] and AMT [161] rely on self-distillation and use attention maps derived from the teacher network to choose the regions to mask. MST chooses to mask non-essential regions only, with a low masking ratio (1/8), while AttMask shows that masking important features at a moderate ratio ($10 - 50\%$) improves the fine-tuning performance. AMT also relies on attention-driven masking, however they use feature maps derived from MAE or SimMIM last layer attention head (thus they do not rely on siamese architecture for training) after a warm-up phase (40 epochs). Like in [160], AMT makes the most informative parts more likely to be masked although there is still a probability that they remain visible. The authors also show that not using middle attention patches increases the performance, while also reducing the training cost. MILAN [162] also proposes a semantic aware sampling by using attention maps derived from CLIP weights [163] (joint text-image SSL pre-training). However, in contrast to AttMask and AMT, a high probability of remaining unmasked is given to highly informative parts. This is motivated by two elements: i)

masking all representative parts of an image leads to very long pre-training, and ii) due to the specific design of their decoder (MAE-like decoder but with frozen representations of the unmasked patches, discussed later), the features extracted by the network need to be informative enough. Indeed, the network should not learn to "hallucinate" objects.

The methods presented so far allow for the decomposition of an image into informative and less informative parts (often foreground/background), and the relationships between those parts (being intra or inter relationships) are learned by the network. What if we further decompose the image by introducing more semantic parts? SemMAE [164] proposes semantic-aware adaptative masking strategy by using segmentation maps. These segmentation maps are learned in a SSL way by solving a reconstruction task where the targets are patches extracted by a pre-trained ViT (e.g., iBOT), and by adding a diversity constraint on the attention maps. The attention maps obtained are then used for segmenting the image into several semantic parts. The semantic-guided masking of SemMAE then consists of progressively masking 75% of each part (intra-part or local feature learning) at the beginning of the training, to masking 75% of the parts (inter-part relationships) at the end of the training process.

Based on all the masking strategies presented so far, the authors seem to agree on the following conclusions: i) a high masking ratio is recommended to ensure meaningful representation learning, and ii) a carefully designed masking strategy (using either attention or semantic maps) further improves the performance. However, it is not clear which parts should be masked. DPPMask [165] may provide an answer that gets everyone on the same page: keep as much representative and diverse information in unmasked patches, while masking at a high ratio (e.g., 75%). Representative and diverse patches are selected using Determinantal Point Processes (DPP), which aims at reducing the semantic change of an image after masking (miss-alignment problem). It consists in computing the distance (using a Gaussian kernel, which depends on the Euclidean distance between the intensity values of the patch pairs) between each patch and selecting those that are dissimilar from a selected subset. Due to the computational complexity resulting from the exact DPP formulation (matrix decomposition), DPPMask proposes a greedy approximation of DPP. DPPMask shows significant improvements over AttMask and SemMAE masking strategies for both MAE and iBOT.

6.2.2 Reconstruction targets

Although masking strategy is very important to improve the performance, there are also many discussions about the choice of the reconstruction targets. First MIM

methods [142, 147, 149] attempt to reconstruct raw pixels, and apply the MSE loss as the reconstruction objective. MAE also considered Principal Component Analysis (PCA) or features extracted from a discrete VAE trained for text-to-image generation [166] as reconstruction targets, however it did not improve the performance. An important limitation with the reconstruction objective is that all reconstructed pixels have the same weight in the loss, although some reconstruction errors may be irrelevant for meaningful feature extraction.

Therefore, some methods propose to adapt the reconstruction target to the downstream objectives. For example, to force the network to focus on shapes rather than texture and rich details, PixMIM [158] filters the high frequencies in the target objective (and thus the network focuses on low frequencies). Ge^2-AE [167] and A^{2}MIM [153] apply the reconstruction loss in both spatial and frequency domains to learn global features. In the same spirit, MaskFeat [168] uses the Histogram of Oriented Gradients (HOG) as a reconstruction target, and justify this choice by the fact that HOG provides local shapes and appearances while being invariant to photometric changes. In the same line, SSM [169] applies different reconstruction losses which introduce some global criteria that do not suppose independence between neighbouring pixels, such as Gradient Magnitude Similarity (GMS) and Structured Similarity Index Measure (SSIM). They also combine corruption prediction and target reconstruction objectives by reconstructing the mask initially applied to the input image.

However, all these methods rely on computationally expensive architectures (compared to instance discrimination) in order to reconstruct the full-resolution (or almost) image, with a decoder that will not be used for fine-tuning. To address this issue, some authors propose to reconstruct some features instead of full-resolution images. In this case, the challenge consists in defining relevant target features. Two solutions have been considered:

- **Features from a pre-trained network** – Several methods like BEiT [146], MaskDistill [170], MILAN [162] and MaskAlign [171], rely on distillation (or self-distillation but using an already pre-trained teacher) from strong unsupervised pre-trained encoders, such as CLIP. However, such a strategy may not be optimal on datasets that present a domain gap (e.g., satellite data) with, for example, CLIP pre-training data.

- **Features obtained via self-distillation** – Another way to obtain target features is by relying on asymmetric siamese networks as with self-distillation methods presented in Section 6.1.1. Such a strategy is adopted by SOTA methods like SplitMask [172], iBOT and I-JEPA [148]. I-JEPA differs from iBOT by the fact that it asks the network to reconstruct various parts only (not the full masked areas) of the masked image given a context. It also

does not optimise a global instance discrimination objective. Nonetheless, the target features obtained using pre-trained weights seem to lead to better representation learning. [173] claims that it is not necessary to carefully choose the target (HOG, MaskFeat or features obtained with MAE/SimMIM etc.) as long as a multi-stage distillation pipeline is used, which leads to dBOT method. However, even with dBOT framework, CLIP pre-trained teacher still leads to better performance than a randomly initialised teacher.

In the literature, many questions have been raised about the design of the decoder in the SSL pre-training phase. Some researchers argue that it is better to use a simple decoder to maximise transfer learning performance [149], while others have observed that a deep and narrow decoder works best [147]. This is one of the open questions in the field of image reconstruction. Indeed, how can we ensure that it is the encoder and not the decoder that learns to extract highly representative information from an image and to disentangle causal factors? MILAN [162] proposes to circumvent this issue by designing a specific decoder that clearly separates the functional roles of the encoder and the decoder. For this purpose, the authors introduce a prompting decoder that takes as input frozen representations of encoded unmasked patches. The latter are therefore used as fixed prompts. However, the ablation study shows that the SOTA performance obtained with MILAN is mainly due to the use of CLIP targets and not to the design of the prompting decoder.

6.3 Conclusion

In this chapter, we have discussed different SSL methods in details. We have grouped them into two categories. The first category involves instance discrimination methods, and includes the following sub-categories: contrastive learning, self-distillation, cross-correlation analysis and clustering-based methods. The second category deals with image modelling methods whose objective is to reconstruct partially corrupted data. This category is further divided according to the types of corruption applied to the input data, as well as the reconstruction objectives considered. We summarise the methods and the categories to which they belong in the Table 6.1.

In the literature, MIM methods are in vogue: when combined with recent Vision Transformers, they can achieve SOTA performance on multiple datasets, outperforming instance discrimination methods, as shown in [147]. The combination of these two SSL categories also looks very promising (e.g., CMAE [174]).

However, most evaluations are carried out on conventional classification or object detection datasets that contain natural images close to those used in the pre-training dataset (e.g., ImageNet). Furthermore, the final task datasets used

Instance discrimination	*Contrastive*		SimCLR, MoCO, PIRL, RELIC, RELICv2, CSI, Spot-the-Difference, SimMoCo, NNCLR
	Self-distillation		BYOL, SimSiam, DINO, DINOv2, iBOT
	Cross-correlation analysis		Barlow Twins, W-MSE, VicReg
	Clustering		DeepCluster, SeLa, SCAN, SwAV, MSN
Image modelling	*Corruption type*	Noise	DAE
		Decolorization	Image colourisation
		Masking	BEiT, MAE, SimMIM, iBOT, I-JEPA, CIM, MST, AttMask, AMT, MILAN, SemMAE, DPP-Mask ...
	Target objective	Raw pixels	MAE, SimMIM
		Image descriptors	PixMIM, Ge^2-AE, A^{2}MIM, MaskFeat, SSM
		Deep features	BEiT, MaskDistill, MILAN, MaskAlign, SplitMask, iBOT, I-JEPA, dBOT

Table 6.1: General overview of SSL methods for image representation learning.

to evaluate SSL methods contain a lot of annotated data. This raises several questions. What if we consider applications from other domains, with different image modalities or objects to be detected, such as the detection of small infrared targets? Is the winning ViT+MIM combo still the most effective in this case? What about pre-training on an in-domain dataset that is not necessarily clean (i.e., presenting temporal redundancy, non object-centric images, or little diversity in the images)? These are open questions in the literature, to which almost no answers have been given. In the next chapter, we will provide some answers by proposing a large number of experiments on several benchmarking datasets, with a focus on small object detection.

Chapter 7

Self-supervised learning methods for small object detection

The methods presented in Chapter 6 were primarily designed for classification tasks, and most of them are benchmarked on classification datasets only. Therefore, their good properties for object detection have only been marginally studied. Although some methods provide promising results on famous object detection datasets like COCO [35] or ADE20K [152], they were not specifically designed for object detection and may thus may appear sub-optimal for this task, and even worse for small object detection. This is especially true for instance discrimination methods that mostly involve inter-image comparisons. Some object-level instance discrimination methods have been proposed in the literature to overcome this problem. Unlike instance discrimination, masked image modelling methods naturally deal with modelling local relationships: neighbour pixels are all the more important to reconstruct masked patches.

In this chapter, we first present some object-level instance discrimination methods that were introduced to improve object detection. We then benchmark several SSL pre-training strategies, originating from different SSL categories, on various benchmarks. We first consider the famous COCO dataset, with a focus on the performance obtained on small objects. We then consider two real-world applications that deal with small object detection, namely vehicle detection from remote sensing data and infrared small target detection.

7.1 Towards object-level representation learning

Some authors propose variants of instance discrimination methods that are well suited for object detection tasks. In the following, we present some improvements of the methods introduced in Section 6 that specifically address dense and local

prediction tasks (e.g., segmentation and object detection, respectively). We will also discuss the pros and cons of the different SSL paradigms with respect to object detection in real-world cases.

7.1.1 Object-level instance discrimination

The instance discrimination methods presented in Section 6.1 assume that the images are semantically consistent. Indeed, the methods are trained on object-centric datasets such as ImageNet, and only inter-image comparisons are performed. However, this hypothesis does not necessarily hold when dealing with dense prediction tasks such as object detection or segmentation. To overcome this issue, two approaches have been investigated: designing data-augmentations at the object or region-level, or applying instance discrimination loss at a local-level (e.g., per pixel). Table 7.1 summaries the different categories and the associated methods. We will discuss both strategies in the following paragraphs.

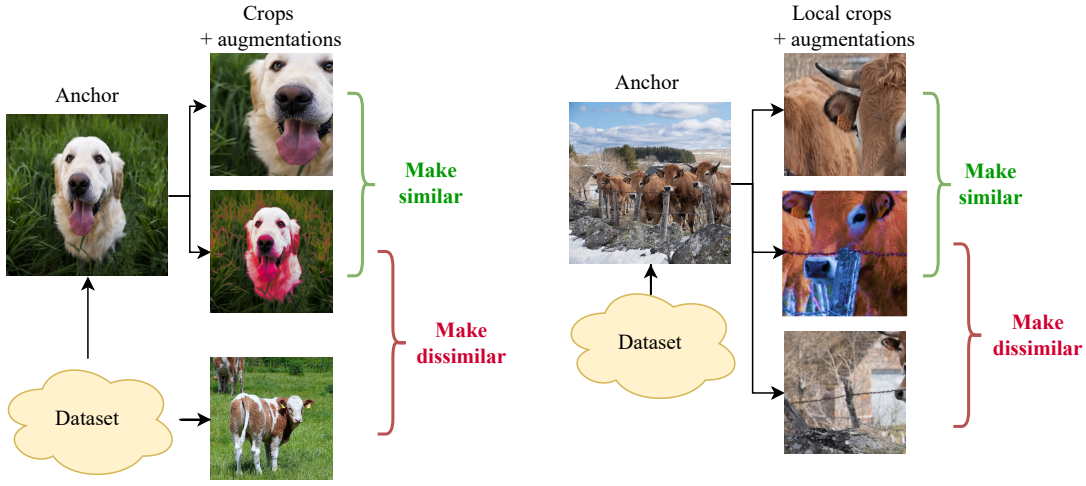
Region-level augmentations		SCRL, **ReSim**, MaskCo, SoCo, CAST, ContrastiveCrop, InsLoc, CP^2, ORL, **Leopart**, InsCon
Dense loss	*Geometric alignment*	VaDeR, PixContrast, **PixPro**, DUPR, InsCon, **Leopart**, LC-Loss, CLOVE
	Feature matching	DenseCL, Self-EMD, VicRegL
	Semantic alignment	**DetCon**, Odin, SetSim

Table 7.1: Taxonomy of object-level instance discrimination methods. The methods we will consider in our experiments are shown in bold.

Region-level augmentations

The first idea for improving SSL for dense or local prediction tasks is to apply instance discrimination loss to local patches in order to perform intra-image instance discrimination, as illustrated on Figure 7.1b). Several strategies have been proposed to ensure semantic consistency between images that form a positive pair.

Spatially Consistent Representation Learning (SCRL) [175] first proposes to randomly select boxes within the intersecting area of the two positive samples and to minimise the similarity between the features predicted by the pooled boxes. Concurrently, [176] proposed a similar approach called ReSim. As shown in Figure 7.2, a sliding window extracts, in each branch, local features within the overlapping area between the two augmented views of the anchor sample (dashed green



a) Inter-image contrastive learning b) Intra-image contrastive learning

Figure 7.1: Inter and intra-image instance discrimination.

area). This creates local positive pairs that represent exactly the same spatial region in the original image (we say that the patches are geometrically aligned). The similarity between the pairs of local patches is thus enforced. Both contrastive (i.e., using negative samples) or similarity (e.g., cosine similarity, positive samples only) losses can be applied to train ReSim. Unlike SCRL, the loss is applied at three different scales in the network, which benefits the detection of objects of different sizes. ReSim also performs inter-image instance discrimination between the two global features (representing the entire positive sample) extracted by the network in order to maintain good performance in classification tasks. MaskCo [177] further introduces the Contrastive Mask Prediction task. It consists in masking one of the local patches (query patch, taken from the first branch), and predicting which augmented view (key views from the second branch) suits the best to fill the masked query patch. Negative key views are introduced by randomly sampling patches from the rest of the dataset, and the contrastive loss is applied to perform the Contrastive Mask Prediction task.

However, SCRL, ReSim and MaskCo assume that all overlapping areas are semantically consistent, which may not be the case on dense visual scenes (e.g., if the size of the overlapping area is too large). To avoid this issue, SoCo [178] relies on the selective search algorithm used in Faster R-CNN to extract semantically consistent sub-regions of an image. Furthermore, CAST [179] introduces saliency random cropping. Saliency maps are learned with Grad-CAM supervision, and their goal is to identify foreground objects (and thus semantically consistent regions) within an anchor image. ContrastiveCrop [180] goes further and proposes

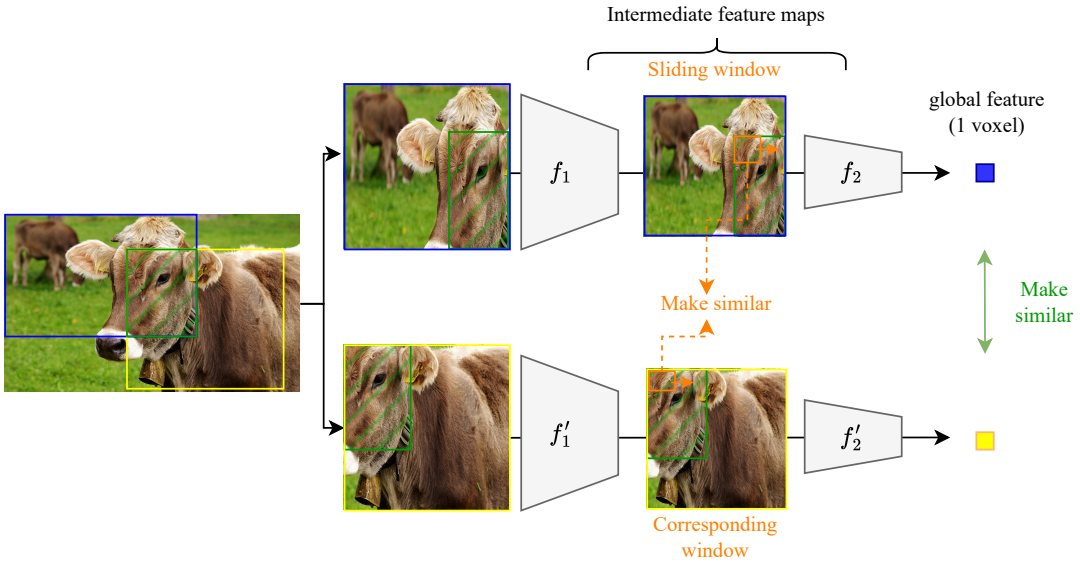


Figure 7.2: Example of object-level instance discrimination pipeline. Here, we represented the ReSim framework, which consists in maximising the similarity between a sliding window in the first branch and its equivalent in the second branch, within an overlapping area.

not only a semantic-aware crop based on the heatmap analysis during the contrastive training, but also a centre-suppressed sampling (i.e., by limiting centre crops and thus avoiding large overlap in the positive pairs) that increases the variance in the crops. Indeed, one issue with random crops is that they may introduce too easy positive pairs, i.e. the positive samples that are very similar. Then, InsLoc [181] and CP^2 [182] introduce background invariance into their crops by copying-pasting foreground images (e.g., crops from ImageNet dataset) on different background images. In their loss, they ensure that the features extracted for the pasted foreground object are similar, regardless of the background.

However, all the methods presented so far rely on intra-image positive pairs, which limits the diversity of information contained in positive pairs. In addition to ContrastiveCrop [180] and their center-suppressed sampling, the method Object-level Representation Learning (ORL) [183] proposed a solution that consists of a three-stage pipeline. First, an instance discrimination method (e.g., BYOL) is trained on an object-centric dataset (e.g., ImageNet) to learn to extract global features. Second, the pre-trained encoder is used to generate local positive pairs. For this purpose, global features are extracted on the target dataset using the pre-trained backbone, and similar images are clustered together using a K-Nearest Neighbours (KNN) algorithm. A selective search algorithm is then used to ex-

tract local regions within the similar images, and positive pairs of local patches are matched using the encoder pre-trained in the first step jointly with a KNN clustering. Third, another instance discrimination method is trained using the newly generated local positive pairs. An alternative is to combine an instance discrimination method based on clustering, such as SwAV, and local augmentations. Leopart [34] builds upon this solution. More specifically, it consists in providing two crops of a foreground object (identified by leveraging ViT attention maps) to an instance discrimination network (e.g., DINO), and then producing patch-level cluster assignments, which are forced to be similar following the online optimisation objective of SwAV [135]. Finally, to improve multi-object detection, InsCon [184] ensures multi-instance consistency by taking as a query sample a multi-instance view containing four images, and as positive samples augmentations of each individual image contained in the query sample.

Dense loss

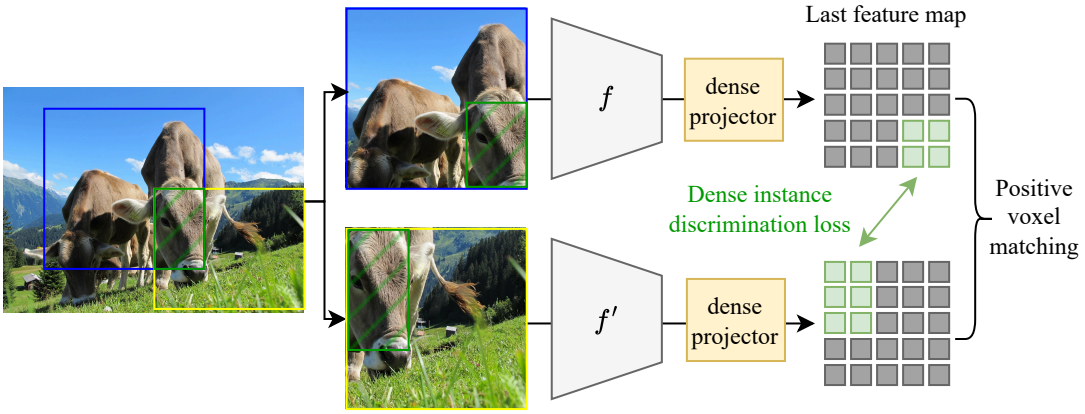


Figure 7.3: Dense instance discrimination loss.

The second idea for improving SSL for dense prediction tasks is to apply an instance discrimination loss at "pixel" level (i.e., each voxel of the last feature map), as illustrated on Figure 7.3. Such a strategy boils down to dividing the image into a grid and taking all (or most) the patches in the grid into account when computing the instance discrimination loss. VaDeR [185] is one of the first frameworks that use such a strategy. To do so, the authors reconstruct the feature map at scale 1/4, and they match the local features extracted from the two positive samples by relying on their geometric correspondence (obtained from the known data-augmentation process). The key to this type of method lies in how the positive voxel are matched, i.e. how the features of different views are aligned. In the literature, several alignment strategies have been proposed:

- **Geometric alignment** – VaDeR [185], PixContrast [186], PixPro [186], DUPR [187], InsCon [184], Leopart [34], LC-Loss [188] and CLOVE [189] assume that the geometric transforms between the positive images are known, and use them to perform spatial alignment. Leopart [34] additionally relies on the attention maps provided by the ViT encoder to focus only on foreground objects in the loss. PixPro further ensures spatial smoothness by propagating the features from similar pixels. CLOVE proposes a similar approach but instead relies on self-attention maps to propagate features.

- **Learned feature matching** – It is not always possible to access geometric correspondences, as for example in the case of temporal positive pairs. Therefore, DenseCL [190], Self-EMD [191] and VicRegL [192] align feature voxels that have a minimal distance between their values. An obvious issue with relying solely on feature alignment is that it assumes that the feature extraction is semantically meaningful, which is not the case at the beginning of the training. On the one hand, DenseCL proposes a warm-up before applying this strategy, although they show that random matching (i.e., not semantically consistent matching) also leads to good performance. On the other hand, VicRegL combines learned feature matching with spatial matching.

- **Semantic alignment** – To ensure semantic consistency between positive pairs of voxels, DetCon [193] estimates pixel categories (pseudo-labels) through unsupervised segmentation masking (using Felzenszwalb-Huttenlocher algorithm [194]). The authors show empirically that more accurate segmentation masks lead to better fine-tuning performance. In the same line, Odin [195] trains an object *discovery* network together with an instance discrimination pipeline. More specifically, the object *discovery* network relies on K-means clustering to cluster the features in the latent space, assuming that each cluster is more likely to represent an object as the training process progresses. Concurrently, SetSim [196] uses attention maps to estimate both positive pixels location and similar sets of pixels, and then computes the similarity between the sets of pixels.

7.1.2 Which SSL paradigm for detecting objects in real-world scenarios?

We have introduced in Chapter 6 some of the most important SSL paradigms as well as some corresponding illustrative methods. In Section 7.1.1, we have provided some examples of how instance discrimination methods can be adapted for object detection, while MIM methods are naturally well-suited for dense prediction tasks.

Although object-level instance discrimination and MIM methods have shown to benefit several dense prediction tasks, it is not clear which paradigm is better suited for object detection. Few studies have attempted to compare the two paradigms:

- [197] benchmarks several object detectors based on ViT encoders, initialised with various pre-trained encoders. They evaluate the fine-tuning performance achieved by supervised pre-training (on ImageNet), MoCov3 [198], BEiT [146] and MAE [147] on the famous COCO dataset. They have found that MAE and BEiT lead to the best results.

- [199] compares contrastive learning and MIM pre-training. The authors observe that contrastive learning models relationships at a global scale (i.e., it learns shapes), whereas MIM is more local (i.e., it has a bias towards textures). This means that later layers (i.e., high-level information) are important for contrastive learning, while early layers (i.e., low-level information) are more important for MIM methods. The authors believe that this is the reason why contrastive learning works best in linear probing on object detection tasks, as it learns to better separate images in the latent space. However, they found that it reduces the diversity of representation, which may impair the performance on dense prediction tasks. Finally, the authors conclude that contrastive learning and MIM methods are complementary, and show that even a basic combination of the two paradigms allows one to benefit from the advantages of both. In the literature, some methods that combine both paradigms are proposed: MSN [136], CMAE [174], Siamese image modelling [200], or CMID [201], which is specifically designed for remote sensing tasks.

- [141] agrees with most of the conclusions drawn by [199]. In their study, the authors compare MIM methods with supervised pre-trained models and MoCov3. They observe that MIM shows a local inductive bias at all layers while supervised pre-trained weights and MoCov3 tend to focus on local details in lower layers and on global details in higher layers. They also show that MIM pre-training brings sufficient diversity to the attention heads, unlike the supervised or contrastive pre-training strategies whose capacity may thus be limited. The authors conclude that coupling MIM methods with ViT encoders should lead to SOTA performance. Indeed, ViT have a very large receptive field, and forcing the ViT to focus on local details helps to optimise it. Regarding the object detection task, they observe that MIM methods help object localisation loss converge faster, while supervised or contrastive pre-trained models benefit object classification more.

- [202] observes the behaviour of several SSL strategies and their fine-tuning performance on different downstream tasks. The authors conclude that al-

though MIM methods often outperform contrastive learning methods on large downstream datasets (e.g., ImageNet), they struggle with data-insufficient datasets. They provide the following explanation: unlike MIM methods, contrastive learning learns more abstract semantics, which helps for unsupervised object recognition. They also conclude that when there is enough data in the final task dataset, low-level features matter the most for achieving good performance on the final task.

Note that the benchmarks are all performed using ViT encoders; the conclusions have not been confirmed on CNN-based backbones such as ResNets. CNN-based encoders are still widely used in many real-world applications and have some advantages, such as faster inference times on small inputs, and a hierarchical architecture that benefits object detection. However, they seem to be less efficient than ViT encoders especially when combined with MIM methods. This may be due to their poor ability to estimate large-scale relationships between image patches. Also, unlike ViT architectures that analyse each patch independently, CNN-based encoders perform convolutions by sliding a window, and thus the receptive field of the convolution can overlap with both masked and unmasked areas. This leads to several issues such as masked pattern vanishing or the disturbance of the distribution of pixel values, as explained in [154]. A^2MIM [153] attempts to solve this issue by replacing the 0-padding with a padding the mean value of the unmasked pixels. ConvMAE [155], MixMAE [156] and SparK [154] introduce the use of partial or sparse convolutions. Specifically, the authors of the SparK [154] paper show that MAE pre-training with a CNN-based encoder can outperform ViT-based MAE pre-training when using sparse convolution and a modern CNN-based encoder, namely ConvX-B [40].

A drawback of these studies is that, in their benchmark, they do not consider the methods based on object-level contrastive learning, which are specifically designed for object detection tasks. Moreover, the benchmarks are primarily performed on the COCO dataset, which is not representative of many real case object detection scenarios. Indeed, in real-world applications, the objects may be very different (e.g., very small objects) and hidden in complex backgrounds. Furthermore, different sensors may be used, for example hyperspectral sensors, so that using the weights of models pre-trained on RGB images is no longer an option. In this case, it is necessary to train the SSL methods on a dataset that shares similar spectral characteristics with the target dataset. The quality of the SSL pre-training (i.e., a pre-training that leads to high fine-tuning performance), as well as the choice of the SSL method, will then heavily depend on the characteristics of the pre-training dataset:

- **Dataset size** – Several papers, including [203], claim that large datasets improve the fine-tuning performance. To tackle the difficulty arising from

small datasets, SimCORE [204] proposes to increase the number of images in the pre-training dataset by automatically selecting open-set images (e.g., images from ImageNet) that are close to the target dataset. However, this strategy cannot be applied to non-natural (i.e., RGB) images. One therefore needs to know if this is a necessary condition for *all* SSL paradigms in order to achieve high performance on the downstream task. [172] explains that, unlike instance discrimination methods, image modelling does not require large training datasets to achieve great fine-tuning performance, especially when pre-training is performed on the target dataset.

- **Diversity and variability** – The authors of the DINOv2 [128] paper have shown that using diverse and curated data from different sources improves the fine-tuning performance for instance discrimination methods. However, it may be difficult to collect enough diverse data in real-world scenarios, especially when the images are extracted from video sequences. Although DINOv2 proposes a pipeline to clean the pre-training dataset from redundant data and increase its diversity, this may lead to a very small pre-training dataset in some cases.

- **Object distribution** – SSL methods are typically pre-trained on object-centric or class-balanced datasets such as ImageNet. Unfortunately, custom pre-training datasets may not share these ideal characteristics: they may suffer from high class imbalance, or they may contain very tiny objects. In this case, which SSL paradigm performs best? Should we adapt the SSL methods (e.g., masking strategy, or local feature matching) accordingly? [150] proposes a random masking strategy that is specifically designed for anomaly detection in remote sensing images. The introduced masks have irregular shapes and sizes that simulate the shapes of anomalies.

- **Semantically consistent scenes** – Two levels of semantic coherence can be discussed: intra-image coherence, and inter-dataset (pre-training and downstream datasets) coherence. The former has been discussed in Section 7.1.1, and obviously MIM methods (using an appropriate masking strategy) and region-level instance discrimination methods are better suited for this case. For the latter, very few studies have been performed, and it is necessary to understand the learning mechanisms of the different SSL methods in order to be able to provide some answers. [203] discusses the semantic coherence requirement for good transfer learning performance when using instance discrimination methods as a pre-training. The authors observe that mid and low-level representations are the most important for achieving good transfer learning performance, and that the semantics (i.e., high-level information) is

not that necessary. Therefore, it may not be necessary to train on semantically similar datasets, but it is important to learn on datasets that have similar low-level statistics. Note that the study was conducted using CNN-based encoders, which naturally present a local-inductive bias. They did not consider ViT encoders or image modelling methods, for which the conclusion may be different.

The choice of the SSL strategy will also depend on the downstream task dataset. There has been very little discussion about the choice of SSL methods according to the final task, and many questions remain without clear answers: are SSL pre-trained weights helpful in the case of frugal training? Do they generalise well to unseen objects or noisy data? In the remainder of the manuscript, we will try to provide some answers with a focus on small object detection in infrared images. For this purpose, we benchmark masked image modeling methods, as well as global and region-level instance discrimination methods on several datasets: i) first, we consider the famous COCO dataset with a focus on the small objects, ii) second, we consider the VEDAI dataset, which tackles the detection of small vehicles in remote sensing images (RGB and infrared), and iii) finally we consider our application of interest, namely infrared small target detection. This will allow us to evaluate all methods on different real-case scenarios, with images captured by different sensors. We will also compare the fine-tuning performance of MIM and object-level instance discrimination on infrared object detection when pre-trained on a custom infrared dataset that has not been curated.

7.2 Benchmark on the COCO dataset

First, we propose to benchmark some self-supervised learning methods in an ideal object detection framework using a classic and widely used dataset from the literature, namely the COCO dataset [35]. It is a large-scale dataset designed for object detection, segmentation, and captioning. The COCO dataset contains more than $200,000$ labelled images covering 80 object categories, including everyday objects such as people, animals, vehicles, food, etc. The scenes are realistic, not object-centred, and the objects are represented in their natural contexts. Various sizes of objects are covered: 41% of small objects (i.e., objects having an area lower than 32^2 pixels), 34% of medium-sized objects (area between 32^2 and 96^2 pixels), and 24% of large objects (area greater than 96^2 pixels). Although what is considered to be "small objects" in the COCO dataset is far from our definition of small targets (i.e., area lower than 9^2 pixels), this dataset remains relevant for assessing the strengths and weaknesses of different self-supervised training strategies on different object sizes.

Method	SSL category	Backbone	#params
DINO [126]	Inst. Discr. (global)	R50	23M
		ViT/S-16	21M
		ViT/B-16	83M
PixPro [186]	Inst. Discr. (local)	R50	23M
ReSim [176]	Inst. Discr. (local)	R50	23M
DetCon [193]	Inst. Discr. (local)	R50	23M
Leopart [34]	Inst. Discr. (local)	ViT/S-16	21M
SparK [154]	MIM	R50	23M
		R200	65M
MAE [147]	MIM	ViT/S-16	21M
		ViT/B-16	83M

Table 7.2: Compared pre-training methods, along with their SSL category, considered backbones and number of parameters in the backbone. R50 stands for ResNet-50 backbone, R200 for ResNet-200, ViT/S-16 for Vision Transformer (ViT) Small version with a patch size of 16, and ViT/B-16 for ViT Base version with a patch size of 16. "Inst. Discr." stands for instance discrimination methods and "MIM" for masked image modeling.

In the literature, although a large number of SSL papers evaluate their methods on the COCO dataset, the fine-tuning set-ups or the evaluation conditions may differ from one paper to another. To ensure a fair comparison, we propose to fine-tune the studied SSL methods ourselves, using training parameters from recent papers that have proven their efficiency. We will benchmark several pre-training strategies belonging to different SSL categories, trained on different backbones (ViTs or ResNets) of different size. We summarise the compared methods with their characteristics in Table 7.2. We divide them into two categories, namely instance discrimination methods ("Inst. Discr." with the local (Section 7.1.1) and global (Section 6.1) distinction, and masked image modelling ("MIM", Section 6.2). In this benchmark we will focus on small object performance, as a prelude to the small object and infrared target detection study that follows in the next section.

Experimental set-up – We consider Mask R-CNN [205] with ResNet-50 (R50), ResNet-200 (R200), ViT-B/16 or ViT-S/16 encoders as our detectors. For the encoder, the pre-trained weights of each SSL method are taken from the Github repository published by the authors of the original papers. The fine-tuning parameters for the ResNet-based encoders (namely R50 and R200) are chosen following SparK's paper [154] recommendations. More specifically, we train the detector

using AdamW optimiser [206] and the 3× schedule (i.e., we trained the network for 3×12 epochs). For the learning rate, since we can only load 36 images on our GPUs, we use the linear scaling rule introduced in [207] to choose an appropriate learning rate. We consider the "Step LR" scheduler, and multiply the learning rate by 0.2 at epochs 3×9 and 3×11. For ViT-based fine-tuning, we follow the training set-up proposed in [208] and scale the learning rate according to our GPU resources based on the linear scaling rule. We fine-tune the neural networks for 50 epochs using AdamW optimiser and CosineLR scheduler.

We evaluate all methods on the "COCO 2017 val" subset, which is a widely used subset of the COCO dataset for evaluating object detectors. We evaluate the box location accuracy of each method using the conventional mean average precision metric $\text{mAP}^{box}_{@0.5:0.95}$ (i.e., the area under the precision-recall curve, averaged over all the object classes and over 10 IoU threshold from 0.5 to 0.95), and do the same for the predicted segmentation mask ($\text{mAP}^{seg}_{@0.5:0.95}$). In order to focus on the detection performance, we will also provide the metrics for box location regardless of the errors made on the classification ($\text{AP}^{box}_{@0.5:0.95}$). We will also focus on small object detection performance by providing this metric for objects that have a spatial extent less than 32×32 pixels ($\text{AP}^{box,S}_{@0.5:0.95}$, $\text{AP}^{box,S}_{@0.3}$). As explained in Chapter 4.2.2, since a small deviation in the box localisation for small objects drastically reduces the IoU between the predicted box and the ground-truth, we introduce more tolerance regarding the localisation errors by lowering the IoU threshold to 30% ($\text{AP}^{box}_{@0.3}$, $\text{AP}^{box,S}_{@0.3}$).

Reproducibility of the results presented in the original papers – Table 7.3 shows the results obtained on COCO val 2017 dataset by the assessed methods. First of all, we would like to make a few comments about the reproducibility of the results presented in the original papers. The results presented in Table 7.3 are obtained after our own fine-tuning, so there may be a difference in performance since we have slightly different fine-tuning conditions. For the methods trained with a ResNet-50 encoder, the results are slightly better than those presented in the original papers. For example, for ReSim, we achieve 44.3% of $\text{mAP}^{box}_{@0.5:0.95}$ with a 3× schedule while the original paper reports a $\text{mAP}^{box}_{@0.5:0.95}$ of 41.9% with a 2× schedule (only 24 epochs for fine-tuning). This difference is probably explained by the choice of a longer schedule, and of a different optimiser: indeed, we considered AdamW optimiser instead of the classical SGD optimiser since it leads to better performance. We have chosen to present the results with more optimal learning parameters and a longer training schedule in order to get closer to the performance obtained with ViT encoders.

For ViT-based fine-tuning, the results are much worse than those reported in the original papers. For example, [147] achieves a $\text{mAP}^{box}_{@0.5:0.95}$ of 50.3% on

	Backbone arch.	$\text{mAP}^{box}_{@0.5:0.95}$	$\text{mAP}^{seg}_{@0.5:0.95}$
Small networks (21-23 M #params.)			
Instance discrimination methods			
DINO	R50	42.8	38.7
PixPro	R50	43.9	39.5
ReSim	R50	44.3	40.1
DetCon	R50	43.6	39.4
DINO	ViT/S-16	<u>46.3</u>	<u>41.1</u>
Leopart	ViT/S-16	**46.5**	**41.5**
MIM methods			
SparK	R50	44.1	39.8
Large networks (>65 M #params.)			
Instance discrimination methods			
DINO	ViT/B-16	<u>47.3</u>	<u>42.0</u>
MIM methods			
SparK	R200	46.7	41.8
MAE	ViT/B-16	**47.8**	**42.6**

Table 7.3: Benchmark on the COCO dataset. For each network size (small or large), best results are in bold and second best results are underlined.

the COCO dataset using MAE pre-trained weight, while we can only achieve a $\text{mAP}^{box}_{@0.5:0.95}$ of 47.8% (−2.5%). This is partly explained by the fact that we considered a shorter fine-tuning schedule (only 50 epochs instead of 100 epochs as in [147]). Moreover, since we did not have access to the same amount of GPU resources as the papers that fine-tuned Mask-RCNN with ViT encoders, we were forced to drastically reduce the size of our batches. Despite adapting the learning rate accordingly, it is likely that the linear scaling rule [207] used for ResNet-based encoders does not directly apply to ViTs. Therefore, it is likely that our training parameters are not optimal. Due to the excessive computation time, the search for optimal training parameters has been set aside, and it must therefore be assumed that there is a slight difference in the results, of about 2% or 3%.

Results – Table 7.3 presents the results (detection *and* classification) obtained on COCOval 2017 dataset, and Table 7.4 presents the detection results only (i.e., we ignore the error made on the classification). Our observations are twofold:

- **The encoder architecture matters more than the SSL strategy** – From Table 7.3, we notice that large networks, especially those based on ViT/B-16 backbone, lead to the best results. For example, the $\text{mAP}^{box}_{@0.5:0.95}$

	Backbone arch.	$AP^{box}_{@0.5:0.95}$	$AP^{box,S}_{@0.5:0.95}$	$AP^{box}_{@0.3}$	$AP^{box,S}_{@0.3}$
Small networks (21-23 M #params.)					
Instance discrimination methods					
DINO	R50	46.9	31.9	77.0	64.0
PixPro	R50	48.2	<u>32.9</u>	77.9	64.0
ReSim	R50	48.6	**33.3**	78.4	65.7
DetCon	R50	47.4	32.1	77.3	64.0
DINO	ViT/S-16	<u>48.8</u>	32.2	<u>79.9</u>	<u>66.9</u>
Leopart	ViT/S-16	**49.0**	32.4	**80.1**	**67.1**
MIM methods					
SparK	R50	48.6	**33.3**	78.0	64.9
Large networks (>65 M #params.)					
Instance discrimination methods					
DINO	ViT/B-16	49.1	32.7	<u>80.2</u>	**68.1**
MIM methods					
SparK	R200	**50.5**	**35.2**	79.4	67.5
MAE	ViT/B-16	<u>50.3</u>	<u>33.4</u>	**80.7**	<u>67.6</u>

Table 7.4: Benchmark on the COCO dataset without classification labels (detection only). For each network size (small or large), the best results are in bold and the second best results are underlined.

is increased by 2.6% when considering a ResNet-200 encoder instead of a ResNet-50 encoder for SparK, and increased by 1% when considering a ViT/B encoder instead of a ViT/S for DINO. Note that the performance gap is narrower for ViT encoders. Moreover, ViT backbones perform significantly better than ResNet backbones. However, the performance gap is reduced if classification errors are ignored, especially when it comes to small objects. Indeed, Table 7.4 shows that SparK initialisation on a ResNet-200 encoder leads to an $AP^{box,S}_{@0.5:0.95}$ that is 1.8% better than MAE initialisation on a ViT/B-16. This means that ResNet encoders are likely to be more prone to classification errors than ViT encoders. Let us now introduce more tolerance towards localisation errors by lowering the IoU threshold to 30%. The results are presented in the last two columns of Table 7.4, and we can notice that ViT encoders outperform ResNet encoders, especially for smaller networks (e.g., +1.4% in $AP^{box,S}_{@0.3}$ when comparing Leopart and ReSim).

- **Introducing locality in the SSL pre-training is important for ResNet-based encoders** – Let us now take a closer look at the performance ob-

tained by each SSL strategy in Table 7.3. Concerning ResNet-50 backbone, it is clear that ReSim outperforms the other pre-training strategies, including the global instance discrimination methods (DINO) or the other local instance discrimination methods such as PixPro or DetCon. SparK (MIM method) leads to competitive performance, while DINO seems to be the worst SSL training strategy for this task. The results seem to be consistent with our intuition: in contrast to global instance discrimination, both local instance discrimination and MIM methods force the networks to model local interactions within the image, which may benefit object detection. For ViT/S-16 and ViT/B-16 backbones, the trends are similar, although the gap in performance is smaller. When looking at the detection performance only on Table 7.4, we can notice that, for ResNet-50 backbones, ReSim and SparK lead to very close results even on small objects, although ReSim is slightly better than SparK when lowering the IoU threshold. The performance gap with DINO remains very large especially for small objects, which suggests that this network will be less suited to our task of small target detection. However, we should note that the difference in performance between the SSL strategies based on ViT encoders is very thin: although local methods (MIM or local instance discrimination methods) seem to perform slightly better in terms of $\mathrm{AP}^{box}_{@0.5:0.95}$, introducing more tolerance towards localisation errors shows that DINO with ViT/S-16 or ViT/B-16 encoder is very competitive on small object detection. Furthermore, DINO with ViT/B-16 encoder leads to the best $\mathrm{AP}^{box,S}_{@0.3}$ score. This suggests that ViT backbones are less sensitive to the pre-training strategy compared to ResNet encoders.

To conclude, this benchmark performed on the COCO dataset suggests that: 1) for ResNet encoders, local instance discrimination and MIM methods are more suited for object detection, especially for small object, 2) ViTs are more relevant for object detection, although they lead to larger localisation errors on small objects, and 3) the fine-tuning performance of Mask R-CNN with a ViT encoder is less sensitive to the SSL pre-training strategy. Note that these conclusions are drawn from a dataset containing fairly large objects, in their everyday life context, and in a sense close to the SSL pre-training data (ImageNet). In the following section, we will investigate whether these conclusions are still valid for datasets that differ from ImageNet or COCO (e.g., in terms of different angles of view, or different spectral bands) and that contain very small objects drowned in different contexts.

7.3 Small vehicle detection from remote sensing imagery

In this section, we challenge some of the previously studied SSL pre-training strategies in several real-world scenarios. We will first evaluate DINO, ReSim, Leopart, SparK and MAE pre-training strategies on the RGB version of the VEDAI dataset (small vehicle detection from remote sensing data, described in Section 2.3.4). We will then study the cross-domain transfer ability of these pre-training strategies to infrared images domain using the infrared version of VEDAI. We will also try to answer the following questions: 1) is it better to perform SSL pre-training on a dataset whose statistics are close to those of the target data? (e.g., infrared dataset, remote sensing data), and 2) which SSL strategy is best for pre-training on an uncleaned dataset (i.e., with high temporal redundancy, low diversity, etc.)? Finally, we will perform a comprehensive analysis of the behaviour of several SSL strategies for the task of infrared small target detection.

7.3.1 Small vehicle detection in RGB images

In this section, we fine-tune a Faster R-CNN on the RGB version of the VEDAI dataset with various encoders initialised with different pre-training strategies (SSL or supervised on ImageNet). The training parameters are those used in Section 7.2, except that we considered a CosineLR scheduler for ResNet-based architectures since it leads to better performance. We considered the same train, validation and test subsets as in Section 5.2.4, as well as the same metrics (object-level F1 score and AP). Since ViT-S/16 weights pre-trained using MAE strategy are not available in the literature, we decided to perform MAE pre-training on ImageNet dataset ourselves. We used the same training parameters as in the original paper but trained the encoder for only 400 epochs (instead of 800) due to time constraints. Note that we tested these pre-trained weights on the COCO dataset, but we did not achieve correct fine-tuning performance on the COCO dataset (conversely to what we will observe on the VEDAI dataset). Therefore, the performance of MAE pre-training on ImageNet with ViT-S/16 is only considered in this section (see Tables 7.5 and 7.6).

Table 7.5 shows the results obtained on VEDAI RGB dataset. First, there is a large gap between the performance obtained using a ResNet-50 and a ViT encoder. In particular, the use of large ViT encoders leads to impressive performance on this dataset. For example, a ViTB/16 encoder can achieve an AP of almost 95% , while ResNet encoders merely reach an AP of 87.7%. Let us now dive into the performance achieved by the different SSL strategies. For the ResNet-50 backbones, ReSim pre-training performs significantly better than DINO

	Backbone arch.	$AP^{box}_{@0.05}$	F1
Small networks (21-23 M #params.)			
Scratch	R50	61.8	62.5
Scratch	ViT/S-16	79.4	72.8
Sup. IN	R50	86.4	82.0
Sup. IN	ViT/S-16	**92.5**	**86.2**
Instance discrimination methods			
DINO	R50	86.1	82.0
ReSim	R50	87.7	84.4
DINO	ViT/S-16	89.7	81.8
Leopart	ViT/S-16	91.0	84.5
MIM methods			
SparK	R50	86.4	83.2
MAE	ViT/S-16	<u>91.8</u>	<u>86.1</u>
Large networks (>65 M #params.)			
Scratch	ViT/B-16	66.7	63.2
Sup. IN	ViT/B-16	**94.9**	**90.0**
Instance discrimination methods			
DINO	ViT/B-16	**94.9**	<u>89.6</u>
MIM methods			
MAE	ViT/B-16	<u>94.1</u>	88.5

Table 7.5: Benchmark of different pre-training methods on the VEDAI dataset (RGB images). For each network size (small or large), the best results are in bold and the second best results are underlined.

and SparK pre-training strategies. For ViT backbones, it is difficult to draw conclusions: MAE seems to benefit the most for small encoder pre-training, while DINO performs slightly better than MAE with a larger encoder. It seems that, for ViT encoders, the fine-tuning performance on the final task is less dependent on the ViT initialisation, which is in line with what was observed on the COCO dataset. Note that the contribution of SSL paradigm for pre-training encoders over supervised ImageNet pre-trained weights is not obvious, since it performs on par with ImageNet supervised pre-training. [202] observed the same trend on data-insufficient downstream tasks.

In the end, it seems that the choice of a good encoder, especially those based on ViT blocks, is more important for the performance of the downstream task than the choice of a good pre-training strategy: indeed, the ImageNet supervised pre-training seems to perform as well as the best SSL pre-training, at least on this

	Backbone arch.	$AP^{box}_{@0.05}$	F1
Small networks (21-23 M #params.)			
Scratch	R50	61.3	60.9
Scratch	ViT/S-16	74.8	71.3
Sup. IN	R50	84.3	80.1
Sup. IN	ViT/S-16	**89.7**	83.0
Instance discrimination methods			
DINO	R50	84.0	79.0
ReSim	R50	85.1	81.6
DINO	ViT/S-16	84.4	78.1
Leopart	ViT/S-16	84.3	78.0
MIM methods			
SparK	R50	81.1	78.4
MAE	ViT/S-16	88.4	**83.7**
Large networks (>65 M #params.)			
Sup. IN	ViT/B-16	91.7	85.9
Instance discrimination methods			
DINO	ViT/B-16	90.7	85.6
MIM methods			
MAE	ViT/B-16	**92.1**	**86.0**

Table 7.6: Benchmark performed on the infrared images of the VEDAI dataset.

dataset. But what if we consider a downstream task dataset whose image statistics are very different from those of ImageNet? This is what we will investigate in the following section using the infrared version of the VEDAI dataset.

7.3.2 Cross-domain transfer ability

Transferring the knowledge learned on RGB data to IR domain – We now evaluate the ability of the different pre-training strategies to transfer to other spectral domains using infrared imagery as a target example. For this purpose, we consider the infrared images of VEDAI dataset and coined this subset of data as VEDAI IR. We fine-tune a Faster R-CNN with different pre-trained encoders in the same way as in the previous sections. Note that these encoders have been pre-trained on RGB images (ImageNet dataset) with different pre-training strategies (supervised or self-supervised). Table 7.6 shows the results obtained on VEDAI IR dataset. We can first notice that there is a large drop in performance for ViT-based instance discrimination pre-training strategies, and they perform even

worse than the ResNet-based pre-trainings (for equivalent network size). Indeed, DINO and Leopart pre-training strategies with ViT/S-16 perform about 5% worse in $AP^{box}_{@0.05}$ when applied to the infrared version of VEDAI, while MAE leads to a decrease of only 2%. The performance gap is less pronounced when it comes to larger networks, and MAE leads to the best performance. For ResNet backbones, ReSim seems to be significantly more robust than any other pre-training strategy, while SparK suffers from a large drop in performance. From these observations we can see that the fine-tuning performance of SSL pre-trained weights varies greatly depending on the encoder architecture considered: MIM methods combined with ViT encoders seem to generalise better to datasets different from the ImageNet dataset, whereas in the case of ResNets, it is the instance discrimination methods that perform best. This may be explained by the fact that MIM methods are very sensitive to image statistics, due to their strong bias towards local details (e.g., textures), and may therefore show a decrease in performance when applied to a different dataset. However, since ViT encoders are better at modelling large-scale dependencies (i.e., they have a bias towards shapes), the combination of ViT encoders and MIM methods compensates for the weakness observed for the latter. Thus the following question arises: can we improve the performance by pre-training on a dataset that has close characteristics to the downstream task dataset? To answer this question, we propose in the next paragraph to perform some SSL pre-training on an infrared dataset, that however is uncleaned (i.e., without removal of redundant images). This will also allow us to assess the degree of generalisability of SSL pre-training to other pre-training databases.

Pre-training on an uncleaned infrared dataset – For this purpose, we collected a large number of infrared images from several publicly available infrared datasets. Table 7.7 summarises the different infrared dataset sources that we merged together in order to obtain a large infrared dataset for SSL pre-training, and we coined the final dataset as **SSL-IR** dataset. The datasets we used to obtain SSL-IR have very different characteristics: they contain different scenes (urban, sky, forest...) captured from various camera viewpoints (drone, car), and with different infrared sensors (thermal infrared, near infrared, etc.). Most of the images are extracted from video sequences, and thus the obtained dataset suffers from low image diversity. We obtain a total of approximately 720k infrared images, which represents about 60% of ImageNet-1k dataset.

We pre-trained ReSim (R50), SparK (R50), Leopart (ViT/S-16) and MAE (ViT/S-16) on the SSL-IR dataset using the pre-training parameters suggested for each method in the original papers. We then fine-tuned a Faster R-CNN on VEDAI IR under the same conditions as before. The results are shown in Table 7.8. ReSim suffers from a huge drop in performance (more than 8% in both AP and F1 score),

Source dataset	Type of data	Nature of data	# images
LSOTB-TIR [209]	drone, car, fixed cameras, urban sky natural scenes, thermal infrared object tracking	video	524k
IRDST [210]	real and simulated data, drone, sky and urban scene, target detection	video	143k
FLIR [211]	car, urban scenes, autonomous driving	video	35k
MFIRST [9]	drone, sky and urban scenes, simulated and real small target detection	single-frame images	10k
ASL-TID [212]	drone, urban scenes, pedestrian detection	video	4k
HIT-UAV [213]	drone, urban scenes, pedestrian detection	video	3k
IRSTD-1k [7]	drone, sky, natural and urban scenes, small target detection	single-frame images	1k
SSL-IR			**720k**

Table 7.7: SSL-IR dataset: data sources and specifications.

while the decrease in performance is limited for SparK and Leopart. Moreover, MAE is particularly robust to training on SSL-IR dataset, since the performance is almost equivalent to the pre-training on ImageNet. Overall, for both ResNet and ViT encoders, MIM-based SSL pre-training is more robust to pre-training on a smaller and less clean dataset than its instance discrimination counterparts.

7.3.3 Conclusion

To conclude this section, let us summarise the main observations:

- Although the different SSL pre-training strategies with ViT-B encoders lead to equivalent performance on the VEDAI RGB dataset (as well as on the COCO dataset), this is no longer the case on the VEDAI IR dataset: MAE combined with ViT-B encoder exhibits a better generalisation ability to statistically different datasets.

- The conclusions are different for ResNet-based SSL pre-trainings: indeed, SparK (MIM method) leads to very weak performance on VEDAI IR dataset. This is explained by the fact that both MIM methods and convolutional networks in general (as opposed to ViT) present a strong bias towards local details.

	Backbone arch.	$AP^{box}_{@0.05}$	F1
Scratch	R50	61.3	60.9
Instance discrimination methods			
ReSim-IR	R50	$76.6^{(-8.5)}$	$72.9^{(-8.7)}$
Leopart-IR	ViT/S-16	$81.6^{(-2.7)}$	$76.7^{(-1.3)}$
MIM methods			
SparK-IR	R50	$77.4^{(-3.7)}$	$75.0^{(-3.4)}$
MAE-IR	ViT/S-16	$\mathbf{88.5}^{(-0.1)}$	$\mathbf{82.8}^{(-0.9)}$

Table 7.8: Benchmark on VEDAI IR with SSL methods pre-trained on SSL-IR dataset. Best results are in bold, and the performance gaps with the respective SSL strategies pre-trained on ImageNet are indicated in the superscript.

- Overall, the SSL pre-training leads to merely better fine-tuning performance than the supervised ImageNet weights, even when considering an important domain gap (RGB versus thermal IR images). The performance of some SSL pre-training strategies are even worse when considering a ViT-S encoder. This could mean that none of the SSL strategies studied in this section is really suited to the downstream task, which has particular characteristics: few fine-tuning data, remote sensing images, etc.

- Pre-training on in-domain data does not improve the performance. This conclusion may not hold if the domain gap is even larger (e.g., medical or astrophysical domains). If an SSL pre-training on a in-domain custom dataset is necessary (e.g., no pre-trained weights are already available), it is preferable to consider MIM-based methods as the SSL strategy.

7.4 SSL and infrared small target detection

We now investigate the benefits of SSL pre-training for our application of interest, namely infrared small target detection. For this purpose, we evaluate different SSL pre-training strategies on the NUAA-SIRST and IRSTD-850 datasets. We then combine them with the method developed in the first part of the thesis, namely the NFA detection head, and evaluate their contribution in a frugal context. Finally, we will try to understand why the performance is degraded when fine-tuning a backbone with SSL initialisation instead of freezing it.

Backbone init.	NUAA-SIRST		IRSTD-850	
	F1	$AP^{box}_{@0.05}$	F1	$AP^{box}_{@0.05}$
Scratch	**97.5**	<u>98.1</u>	85.3	91.3
Sup. IN	96.5	97.1	87.9	92.7
+ Freeze	$94.9^{(-1.6)}$	$97.7^{(+0.6)}$	<u>89.1</u>$^{(+1.2)}$	**94.1**$^{(+1.4)}$
Instance discrimination methods				
DINO	<u>97.4</u>	97.6	**89.4**	94.2
+ Freeze	$97.3^{(-0.1)}$	**98.2**$^{(+0.6)}$	$87.8^{(-1.6)}$	<u>93.9</u>$^{(-0.9)}$
ReSim	94.7	97.0	87.9	92.8
+ Freeze	$96.3^{(+1.6)}$	<u>98.1</u>$^{(+1.1)}$	$86.3^{(-1.6)}$	$92.3^{(-0.5)}$
ReSim-IR	95.4	97.4	86.0	92.0
+ Freeze	$95.2^{(-0.2)}$	$97.0^{(-0.4)}$	$86.5^{(+0.5)}$	$90.7^{(-1.3)}$
MIM methods				
SparK	96.5	<u>98.1</u>	86.6	91.6
+ Freeze	$94.7^{(-1.8)}$	$97.6^{(-0.5)}$	$88.4^{(+1.8)}$	$93.1^{(+1.5)}$
SparK-IR	95.4	97.4	88.1	93.6
+ Freeze	$95.8^{(+0.4)}$	$97.3^{(-0.1)}$	<u>89.1</u>$^{(+1.0)}$	$93.3^{(-0.3)}$

Table 7.9: Results obtained on SIRST and IRSTD-850 datasets using a YOLO-R50 with different backbone initialisations. For each pre-training strategy, we also show the results when freezing the backbone ("+ Freeze" row), and the performance gap is indicated in the superscript. The best results are in bold, and the second best results are underlined.

7.4.1 Contribution of the SSL in a data-sufficient regime

We first evaluate the benefits of several ResNet pre-training strategies on NUAA-SIRST and IRSTD-850 datasets. More specifically, we evaluate the benefits of ImageNet pre-training (Sup. IN), some contrastive methods (DINO, ReSim pre-trained on ImageNet as well as on SSL-IR), and SparK pre-trained on ImageNet and SSL-IR. In our experiments, we have found that Faster RCNN leads to particularly poor performance on NUAA-SIRST dataset. Thus, we rather consider YOLO architectures. We can easily substitute the YOLOv7-tiny backbone with a ResNet-50, and coin this new version of YOLO as YOLO-R50. Since no implementation of YOLO with ViT backbone was available in the literature, we only assess ResNet-based SSL pre-training in the following. The training set-up is the same as in Section 5.2. In the tables, the presented results have been averaged over three distinct training sessions.

Table 7.9 presents the results obtained when fine-tuning the entire YOLO-R50 network (no freeze) on NUAA-SIRST and IRSTD-850 with different backbone

initialisations. First of all, we can notice that all pre-training strategies downgrade the performance on SIRST. This may be explained by the fact that 1) NUAA-SIRST dataset contains enough samples and is quite easy for the task of infrared small target detection (since training from scratch reaches 97.5% of F1 score), and 2) that the fine-tuning strategy may not be adequate (this will be addressed in the next paragraph). Given these results, it is not reasonable to draw any conclusions about the contribution of the SSL pre-training strategies for infrared small target detection. Let us now take a look at the performance achieved on IRSTD-850 dataset. We can note that, in this case, the SSL pre-training benefits the fine-tuning performance, probably because the dataset presents more challenging scenes compared to SIRST dataset. More specifically, instance discrimination methods like DINO perform well, while MIM methods seem to need a fine-tuning on an in-domain dataset: indeed, SparK-IR outperforms outperforms SparK by 1.5% in F1 score. As expected, pre-training an instance discrimination method like ReSim on an in-domain dataset leads to a decrease in performance as the hypothesis made on the semantic consistency between two positive images does not hold on our custom SSL-IR dataset. Thus, the features extracted by ReSim-IR are less informative.

In the literature, encoders pre-trained with SSL methods are evaluated on the downstream tasks after fine-tuning the entire classification, detection or segmentation network. However, as explained in [214] and observed in Table 7.9, such a fine-tuning strategy may not be suitable for dense prediction tasks. This can be explained by the fact that a complex, randomly initialised detection or segmentation head has to be added on top of the encoder, and the backpropagation of these random weights can "break" the knowledge learned during the SSL pre-training of the encoder. To further improve the transfer learning performance on IRSTD and to investigate the benefits of each SSL method without the confounding effects of fine-tuning, we propose to freeze the backbone layers (i.e., the ResNet layers in YOLO-R50) and to fine-tune only the rest of the neural network (i.e., the YOLOv7-tiny detection head), as in [214]. The results are presented in the "+ Freeze" rows of Table 7.9, and we make the following two observations:

- **Global instance discrimination methods generalise well to IRSTD tasks** – Indeed, in both settings, DINO initialisation leads to high final performance on both datasets. MIM methods seem more adequate for IRSTD-850 dataset, especially when the pre-training is performed on an in-domain dataset. However, it can be noticed that ImageNet pre-training performs on par with SSL pre-trainings. This suggests that the pre-training strategy does not really matter in a data-sufficient context, although pre-training on an in-domain dataset may be helpful.

- **An appropriate fine-tuning strategy should be considered** – Let us now compare the performance of fine-tuning with and without freezing the

Backbone init.	NUAA-SIRST		IRSTD-850	
	F1	$AP^{box}_{@0.05}$	**F1**	$AP^{box}_{@0.05}$
Scratch	$96.9^{(-0.6)}$	$98.3^{(+0.2)}$	$\underline{91.1}^{(+5.8)}$	$93.9^{(+2.6)}$
Sup. IN + Freeze	$\underline{98.1}^{(+3.2)}$	$\mathbf{98.6}^{(+0.9)}$	$89.7^{(+2.8)}$	$\mathbf{95.3}^{(+3.1)}$
Instance discrimination methods				
DINO + Freeze	$97.5^{(+0.2)}$	$\mathbf{98.6}^{(+0.4)}$	$90.8^{(+3.0)}$	$\underline{94.9}^{(+1.6)}$
ReSim + Freeze	$\mathbf{99.1}^{(+2.8)}$	$\mathbf{98.6}^{(+0.5)}$	$89.8^{(+3.5)}$	$\mathbf{95.3}^{(+3.0)}$
ReSim-IR + Freeze	$\underline{98.1}^{(+2.9)}$	$\underline{98.5}^{(+1.5)}$	$88.6^{(+2.1)}$	$93.0^{(+2.3)}$
MIM methods				
SparK + Freeze	$97.8^{(+3.1)}$	$\underline{98.5}^{(+0.9)}$	$\underline{91.1}^{(+2.7)}$	$94.8^{(+1.8)}$
SparK-IR + Freeze	$97.4^{(+1.6)}$	$97.6^{(+0.3)}$	$\mathbf{91.3}^{(+2.2)}$	$\underline{94.9}^{(+1.6)}$

Table 7.10: Results obtained on SIRST and IRSTD-850 datasets using a YOLO-R50+NFA with different backbone initialisations. We only provide the results of training strategies where the backbone is frozen, as this gave the best results. For each pre-training strategy, the performance gap with the YOLO-R50 architecture (i.e., without the NFA detection head) and its respective frozen backbone initialisation is indicated in the superscript. The best results are in bold, and the second best results are underlined.

backbone. In some cases, freezing the backbone improves the final performance, sometimes by a large margin: for example, on IRSTD-850 dataset, the F1 score is increased by 1.8% when freezing the backbone initialised with SparK-IR weights. This phenomenon has been observed and analysed by [214]. Specifically, they have found that a long-training schedule (i.e., complete training with a large number of epochs) moves the backbone representation away from the initial one. They have shown that a simple backbone freezing allows one to preserve the knowledge brought by a good backbone initialisation and improves the final performance, not only in short but also in long-training schedules. This improvement is even greater when using an advanced detection head (as it is the case with YOLOv7 detection head). Although freezing the backbone entirely may not be an optimal strategy, it suggests that designing a good fine-tuning strategy is essential to reap the full benefits of SSL pre-training in a data-sufficient context.

7.4.2 Combining SSL and NFA detection head

In this section, we evaluate the benefits of combining our NFA detection head with the different SSL initialisation studied earlier. The training set-up is the same as before, and the results presented in the tables have been averaged over

three distinct training sessions. Table 7.10 presents the performance achieved by the assessed methods on NUAA-SIRST and IRSTD-850 datasets, along with the performance gap with the YOLO-R50 architecture in the superscript. Note that we only consider the transfer learning strategy where the backbone is frozen, as this gave the best results. The conclusions on SIRST dataset are quite different from those of Section 7.4.1. Indeed, although DINO backbone initialisation leads to very competitive results on both SIRST and IRSTD-850 datasets, the best performance is achieved by ReSim on the SIRST dataset and by MIM methods on IRSTD-850. More specifically, ReSim achieves more than 99% of F1 score, outperforming all other methods evaluated so far (including YOLO-R50, or SOTA segmentation networks) with a wide margin. ReSim-IR and supervised ImageNet pre-training weights also lead to very competitive performance. As observed for YOLO-R50, it is the instance discrimination methods that seem to benefit the detection performance on SIRST dataset the most. Regarding IRSTD-850 dataset, pre-training SparK on SSL-IR dataset leads to impressive results, outperforming DINO initialisation, and leading to competitive performance compared to DNANet. The trend remains similar to what was observed in Section 7.4.1: local instance discrimination methods are less efficient on IRSTD-850 dataset.

7.4.3 SSL and frugal setting

Let us now evaluate the SSL pre-training under more challenging conditions, namely 25-shot and 35-shot training on SIRST. The results for the YOLO-R50 baseline and YOLO-R50+NFA are presented on Table 7.11, and they have been averaged over three distinct training sessions. Note that here we fine-tune the entire network (including the backbone) since we deal with very challenging training conditions (and freezing the backbone did not improve the performance). Our observations are threefold:

- **The NFA detection head contributes the most to good performance in frugal setting** – Considering 25-shots, YOLO-R50+NFA achieves a F1 score of 91.9% with only 10% of SIRST dataset, while YOLO-R50 with the best backbone initialisation leads to a F1 score of only 51.6%. Furthermore, combining YOLO-R50+NFA with DINO initialisation allows us to reach a F1 score of 97.1%, which is very close to the performance achieved when training on the entire dataset. YOLO-R50+NFA also outperforms YOLO-R50 no matter the considered backbone initialisation in the 35-shot setting.

- **MIM methods perform poorly in a few-shot setting** – From Table 7.11, it is clear that all SSL initialisations benefit the final task performance. For example, the F1 score is multiplied by a factor of 2 when using

Backbone init.	NFA	25-shots	35-shots
Scratch		26.1	38.4
Scratch	✓	91.9	94.8
Instance discrimination methods			
DINO		51.6	92.7
ReSim		43.6	90.4
ReSim-IR		36.4	76.4
DINO	✓	**97.1**	<u>96.6</u>
ReSim	✓	<u>95.4</u>	**97.2**
ReSim-IR	✓	93.0	95.9
MIM methods			
SparK		16.3	56.0
SparK-IR		13.7	44.5
SparK	✓	93.5	95.1
SparK-IR	✓	93.0	94.8

Table 7.11: F1 score achieved by YOLO-R50 with different backbone initialisations in 25 and 35-shot settings on NUAA-SIRST. The contribution of the NFA detection head is also presented. The best results are in bold and the second best results are underlined.

an instance discrimination method as the backbone initialisation of YOLO-R50 in the 35-shot setting. DINO and ReSim both benefit greatly few-shot training, no matter the detector (YOLO-R50 or YOLO-R50+NFA). This is not the case for MIM methods, which perform no better than random initialisation. Once more, this is consistent with the conclusions drawn by [202]: MIM methods tend to perform poorly in data-insufficient setting compared to instance discrimination methods.

- **Instance discrimination methods are better suited in a frugal context** – Although ReSim-IR performs worse than ReSim, it still contributes to better performance compared to random weights or MIM methods in a frugal setting. Therefore, it is preferable to consider instance discrimination methods when dealing with very little data in the final task, even when a custom SSL pre-training on in-domain data is needed.

Backbone init.	NUAA-SIRST	
	F1	$AP^{box}_{@0.05}$
Scratch	<u>97.5</u>	98.1
+ NFA	96.9	98.3
Sup. IN	$96.5^{(-1.0)}$	$97.1^{(-1.0)}$
+ NFA	<u>97.5</u>$^{(+0.6)}$	<u>98.4</u>$^{(+0.1)}$
Instance discrimination methods		
DINO	$97.4^{(-0.1)}$	$97.6^{(-0.5)}$
+ NFA	$97.0^{(+0.1)}$	<u>98.4</u>$^{(+0.1)}$
ReSim	$94.7^{(-2.8)}$	$97.0^{(-1.1)}$
+ NFA	$96.8^{(-0.1)}$	$98.2^{(-0.1)}$
ReSim-IR	$95.4^{(-2.1)}$	$97.4^{(-0.7)}$
+ NFA	$95.9^{(-1.0)}$	$98.0^{(-0.3)}$
MIM methods		
SparK	$96.5^{(-1.0)}$	$98.1^{(+0.0)}$
+ NFA	$97.3^{(+0.4)}$	$98.3^{(+0.0)}$
SparK-IR	$95.4^{(-1.1)}$	$97.4^{(-0.7)}$
+ NFA	**97.8**$^{(+0.9)}$	**98.5**$^{(+0.2)}$

Table 7.12: Results obtained on SIRST dataset when fine-tuning the entire YOLO-R50 and YOLO-R50+NFA architectures with different backbone initialisations. For each method, we indicate the performance gap with the training from scratch (for each architecture respectively). The best results are in bold, and the second best results are underlined.

7.4.4 What happens when fine-tuning with a long-training schedule?

In the previous subsections, we have noticed that fine-tuning the backbone can impair the final performance when using pre-trained weights for the baseline YOLO-R50. Table 7.12 adds the results obtained when fine-tuning the entire YOLO-R50+NFA with the different backbone initialisation. Unlike with YOLO-R50, we can see that the performance of YOLO-R50+NFA is less affected by the fine-tuning step. To investigate this, let us analyse the feature representation learned by each backbone, depending on its initialisation and the architecture that was considered (with or without the NFA detection head). For this purpose, we compute the layer-wise representation self-similarity for each fine-tuned backbone, depending on its initialisation. The CKA metric, described in Appendix A, is used to compute the similarity between two feature maps, and we consider the last layers of

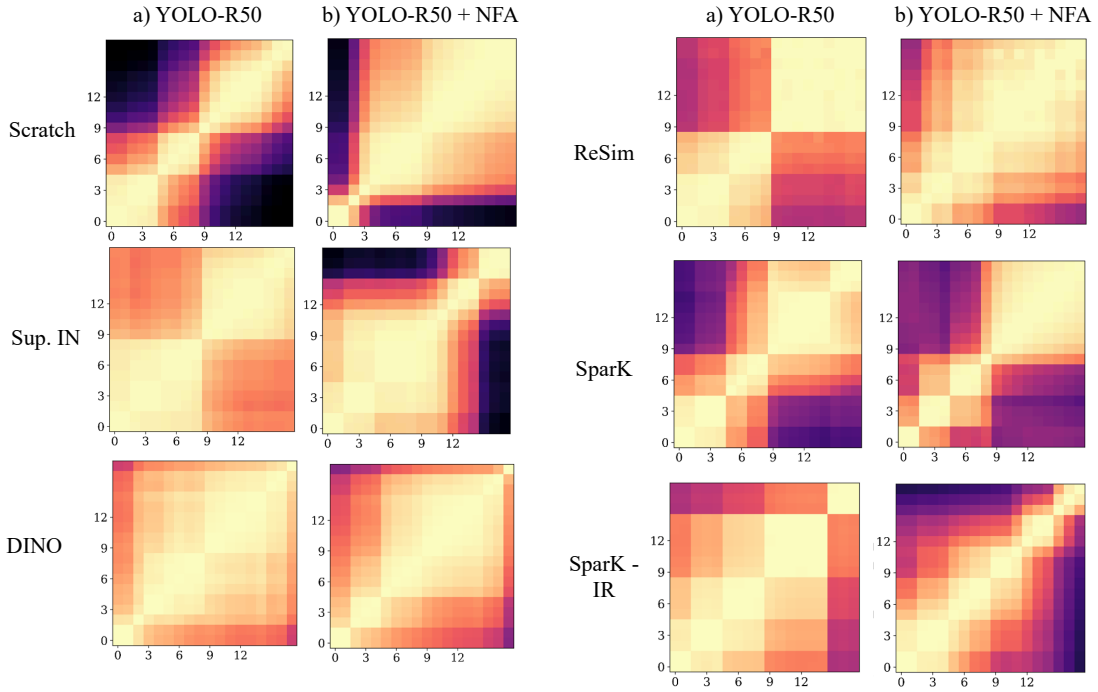


Figure 7.4: CKA maps computed for several fine-tuned networks, depending on the architecture (YOLO-R50 or YOLO-R50+NFA).

each convolution block of a ResNet backbone in our study (which makes a total of 18 layers). The layers are ranked from 0 to 18, and ordered from the lowest level (close to the input) to the highest level (more abstract layers). Figure 7.4 presents the CKA maps computed for each fine-tuned backbone (i.e., we compute the self-similarity for each model) over SIRST test set. Note that the higher the similarity between the representations provided by two layers is, the higher the CKA score will be. Since we compute the self-similarity within the same network, the diagonal terms are equal to 1. From Figure 7.4, we can notice that, except for the weights trained from scratch, the layers of the YOLO-R50 are very self-similar. Indeed, we distinguish very large blocks: this means that the layers that lead to these large blocks extract the same information. According to [215], this could be a consequence of an over-parametrisation of the network for this task. In other words, removing layers from this block would merely affect the final performance. Furthermore, we can observe that adding the NFA criterion into the detection head leads to the emergence of smaller blocks, meaning that there is a higher representation diversity. This is especially the case for SparK-IR weights, which indeed lead to the best performance on SIRST dataset. Having a CKA maps with smaller blocks could thus go hand in hand with better performance.

Although YOLO-R50+NFA leads to more diverse representations in the latent

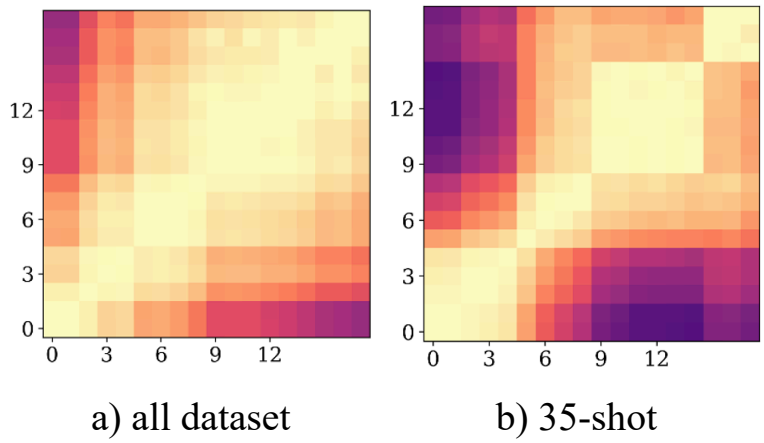


Figure 7.5: CKA maps for ReSim + NFA in a) data-sufficient and b) frugal (35-shot) settings.

space compared to the baseline, the representations learned with instance discrimination initialisation still lack diversity, and they lead to weak performance in Table 7.12. Furthermore, if we look closely at the performance obtained in the 35-shot setting (Table 7.11), we can see that ReSim with YOLO-R50+NFA performs better in this setting than in the data-sufficient context (as shown in Table 7.9). To investigate this, we compute the CKA map of ReSim backbone (self-similarity) trained in the 35-shot setting. Figure 7.5 compares the self-similarity of ReSim in a data-sufficient context (Figure 7.5a) and in the 35-shot setting (Figure 7.5a). It is clear that we can see the emergence of smaller blocks in the frugal setting, meaning that there is a higher diversity in the learned representations. It seems that a long-training schedule, or training in a data-sufficient setting, leads to a representation collapse in the network. In this case, a correct fine-tuning strategy should be designed in order to avoid such phenomenon.

7.5 Conclusion

In this chapter, we have evaluated different SSL pre-training strategies on several datasets. In particular, we have considered global and local instance discrimination methods, as well as MIM methods. We first evaluated them on the COCO dataset, which is statistically close to the pre-training dataset (namely ImageNet dataset), and we focus on the performance obtained on small object detection. We then considered the VEDAI dataset and two small target detection datasets to test the limits of these methods. The conclusions for the COCO and VEDAI datasets are fairly similar: the MIM and ViT combo gives the best performance,

which is consistent with what is announced in the literature. For convolutional networks, ReSim (local discrimination instance) seems to be the most robust. On the other hand, MIM methods (whether combined with ViT or ResNet encoder) are to be preferred when pre-training on a custom in-domain (with respect to the downstream task) and uncleaned dataset.

For small target datasets, the conclusions are less firm. The SSL paradigm seems to contribute only when the dataset is difficult, or under complicated learning conditions (e.g., few shot setting). On the SIRST dataset, the SSL degrades performance if the fine-tuning strategy is not appropriate. Freezing the backbone weights helps to limit this drop in performance. The fine-tuning strategy must therefore be chosen meticulously in the future. Overall, global instance discrimination methods (e.g., DINO) seem to be appropriate in most cases when considering convolutional encoders. For local SSL pre-training strategies such as SparK or ReSim, the conclusions are different: ReSim seems to be better suited for SIRST dataset, especially when combined with the NFA detection head, while SparK trained on an SSL-IR dataset gives excellent performance on IRSTD-850. Furthermore, in contrast to the VEDAI dataset, we have seen here the benefits of training on a custom IR dataset: SparK-IR performs much better than SparK on IRSTD-850. This is not the case with ReSim. This confirms that MIM methods are more appropriate for pre-training on an uncleaned, in-domain datasets.

It should be noted that all these conclusions have been drawn from detection networks based on convolutional encoders only. Since Faster R-CNN with ViT encoder performed too poorly on SIRST datasets in our experiments, we were unable to evaluate the contribution of ViT. The conclusions may have been different, especially for MIM methods, when considering a ViT encoder. A hierarchical ViT-based architecture (e.g., Swin Transformers) that limits the information loss on small objects should be considered in the future in order to complete this benchmark on small targets. We will discuss this option in the general conclusion of the manuscript.

Finally, in Section 7.4.2, we have combined the contributions of the both Parts I and II by initialising the encoder of our best detector, namely YOLO + NFA, with SSL pre-trained weights. Mixing both NFA and SSL paradigms effectively improves previous results, leading to new SOTA performance for infrared small target detection.

Chapter 8

Conclusion and perspectives

8.1 Conclusion

In this thesis, we have explored two approaches to enhance infrared small target detection. First, we aimed to improve the detector performance by integrating an *a contrario* decision criterion into the training process of both segmentation and detection networks. Second, we examined the advantages of self-supervised pre-training for IRSTD. To this end, we conducted a survey on SSL methods for image representation learning, with an emphasis on some methods adapted for small object detection. We then benchmarked several SSL strategies across different datasets, including IRSTD datasets. Our findings are summarised in the following two sections.

8.1.1 New SOTA results for IRSTD

Both the *a contrario* and self-supervised paradigms have led to impressive results for IRSTD. In Table 8.1, we present the results obtained on the SIRST and IRSTD-850 datasets, with the performance of the best SOTA baseline, DNANet, highlighted in italics. The table includes only the top-performing methods developed in Chapters 4, 5 and 7, and the performance gap with DNANet is indicated in the superscript. The *a contrario* paradigm clearly benefits both segmentation (with the DNIM backbone) and object detectors (using YOLOv7-tiny and YOLO-R50) for IRSTD. Additionally, applying the NFA test on the top of DNIM backbone or YOLO detectors sets new SOTA results on the SIRST dataset. Combining SSL with the *a contrario* approach further boosts performance on SIRST dataset: YOLO-R50 + $\text{NFA}_{\mathcal{N}}$ initialised with ReSim weights outperforms DNANet with a large margin (+2%), achieving more than 99% of F1 score. Initialising YOLO-R50 + $\text{NFA}_{\mathcal{N}}$ with SparK weights pre-trained on SSL-IR dataset also leads to SOTA performance on the IRSTD-850 dataset. This highlights the potential of lever-

Backbone init.	NUAA-SIRST		IRSTD-850	
	F1	**AP**	**F1**	**AP**
SOTA IRSTD baselines				
ACM	95.4	95.2	62.1	48.4
LSPM	92.9	90.2	54.9	51.5
AGPCNet	93.8	92.2	88.1	92.3
MTU-Net	93.8	97.2	86.8	89.0
DNANet	*97.1*	*98.4*	*91.4*	*92.4*
DNIM and YOLO baselines				
DNIM	95.8	96.2	89.0	89.9
YOLOv7-tiny	96.5	97.8	82.2	85.0
YOLO-R50	97.5	98.1	82.3	84.3
Our methods				
DNIM+NFA$_{\mathcal{N}}$ (Ch. 4)	97.6$^{(+0.5)}$	98.4$^{(+0.0)}$	91.3$^{(-0.1)}$	94.2$^{(+1.8)}$
YOLOv7-tiny+NFA$_{\mathcal{N}}$ (Ch. 5)	97.6$^{(+0.5)}$	98.3$^{(-0.1)}$	90.1$^{(-1.3)}$	94.1$^{(+1.7)}$
YOLO-R50+NFA$_{\mathcal{N}}$+ReSim (Ch. 7)	**99.1**$^{(+2.0)}$	**98.6**$^{(+0.2)}$	89.8$^{(-1.6)}$	**95.3**$^{(+2.9)}$
YOLO-R50+NFA$_{\mathcal{N}}$+SparK-IR (Ch. 7)	97.4$^{(+0.3)}$	97.6$^{(-0.8)}$	91.3$^{(-0.1)}$	<u>94.9</u>$^{(+2.5)}$
YOLOv7-tiny-1 scale+NWD+NFA$_{\mathcal{N}}$	97.5$^{(+0.4)}$	98.4$^{(+0.0)}$	**92.5**$^{(+1.1)}$	**95.3**$^{(+2.9)}$

Table 8.1: Overview of the performance obtained by SOTA IRSTD methods, DNIM and YOLO baselines as well as our methods on SIRST and IRSTD-850 datasets. The best performance are given in bold. The results of the SOTA IRSTD method, DNANet, are indicated in italics, and the performance gaps between our methods and DNANet are provided in the superscript.

aging large unlabelled in-domain datasets for SSL pre-training of the encoder to improve downstream task performance. Last but not least, considering a YOLO baseline that is specifically tailored for small object detection (e.g., by removing low resolution scales or by introducing a Gaussian prior) further improves the performance, surpassing DNANet with a large margin on IRSTD-850 dataset (+1.1% in F1 score and +2.9% in AP compared to DNANet).

Nevertheless, the most impressive asset of the methods we developed is their significant robustness under challenging training conditions, such as few-shot training. Table 8.2 compares the results obtained by the baselines and our methods in a 25-shot setting on the SIRST dataset. The contribution of the NFA head is particularly impressive on YOLO baselines, with our best method, namely YOLO+NFA initialised with ReSim weights, improving the baseline by approximately 70%. Moreover, all of our methods outperform DNANet by a wide margin. Notably, integrating our NFA detection head into the YOLO backbone allows us to achieve

Method	25-shots	
	F1	**AP**
SOTA IRSTD baselines		
DNANet	*73.1*	*63.7*
DNIM and YOLO baselines		
DNIM	83.4	83.6
YOLOv7-tiny	21.8	15.0
YOLO-R50	26.1	23.1
Our methods		
DNIM+NFA$_{\mathcal{N}}$ (Ch. 4)	90.9	93.1
YOLOv7-tiny+NFA$_{\mathcal{N}}$ (Ch. 5)	93.6	95.0
YOLO-R50+NFA$_{\mathcal{N}}$+ReSim (Ch. 7)	**95.4**	**96.6**
YOLO-R50+NFA$_{\mathcal{N}}$+SparK-IR (Ch. 7)	93.5	95.1

Table 8.2: Results achieved in a 25-shot setting on NUAA-SIRST. Best results are in bold.

a F1 score that is nearly as high as that obtained with the full SIRST dataset, even when using only 10% of the training data. This highlights the robustness of our methods in difficult training scenarios. Additionally, we have shown in Chapter 4 (Section 4.2.3) that the *a contrario* paradigm is beneficial for domain adaptation (i.e., training on one dataset and transferring knowledge without fine-tuning to a different dataset) and for making inferences on noisy data.

8.1.2 Roadmap for selecting SSL methods for IRSTD

In the second part of this manuscript, we evaluated several SSL methods across different benchmarks. Specifically, we compared global and local instance discrimination techniques, as well as masked image modelling methods, and drew key conclusions on which method is better suited for IRSTD tasks and under what conditions. In this paragraph, in the light of the insights presented in Chapter 7, we aim to provide a roadmap to guide the future practitioners in selecting an appropriate SSL strategy based on various parameters or conditions. The roadmap is illustrated on Figure 8.1.

The first question to consider is whether SSL pre-training is necessary. Indeed, we have observed that pre-training on an IR dataset improves the detection performance on IRSTD-850, while it does not necessarily lead to better performance on SIRST dataset compared to using SSL weights pre-trained on RGB datasets like ImageNet. This suggests that SSL pre-training on custom in-domain dataset is all the more important when there is a significant domain gap between the available

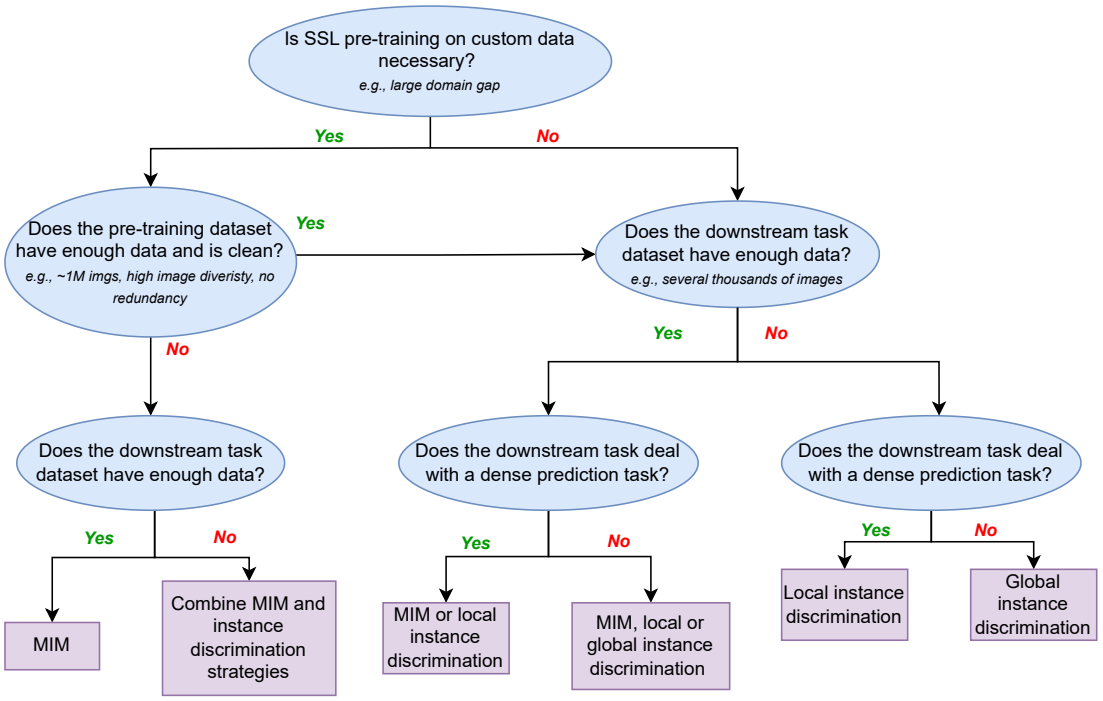


Figure 8.1: Roadmap for selecting SSL strategies.

SSL pre-trained weights (such as those pre-trained on RGB images) and the downstream task (like segmentation specific images such as medical or remote sensing ones).

- In the case where custom pre-training is *not* needed, the choice of suitable SSL pre-trained weights will depend on the amount of training data available for the downstream task. As observed in Section 7.4.3, MIM methods perform poorly in few-shot settings. Therefore, we will prefer instance discrimination methods over MIM methods when dealing with a frugal downstream task dataset. Furthermore, if the final task involves dense or local prediction, local methods (local instance discrimination, or MIM methods if enough training data is available) may be preferred over global ones.

- If SSL pre-training is necessary but the pre-training dataset is limited or uncleaned, instance discrimination methods should be avoided. MIM methods are better suited for small, low diverse and unclean pre-training datasets. However, since they tend to underperform in frugal settings, a combination of MIM and instance discrimination strategies, as done in [127, 201], should be considered when dealing with frugal downstream task datasets.

A factor that is not fully addressed in this roadmap is the use of a ViT back-

bone. Combining MIM with a ViT encoder appears to be a winning strategy, as demonstrated in the benchmarks conducted on the COCO and VEDAI datasets. However, we were unable to confirm whether this conclusion applies to small target detection. If it does, the conclusions regarding MIM methods in frugal contexts would need to be re-evaluated.

The research work conducted during this thesis has resulted in several publications, listed in Appendix B. Due to the thesis timing, the papers already published focus mainly on the first part of the manuscript. We first introduced a proof of concept demonstrating the benefits of the NFA test as a post-processing step in [216] (corresponding to Chapter 3). Following this, we published an article detailing the integration of an *a contrario* criterion into the training loop of a segmentation network in [94] (corresponding to Chapter 4). Lastly, we presented our work at the ICASSP conference, where we discussed the integration of an object-level *a contrario* criterion into a YOLO detection head [217] (Chapter 5). For Part II, two articles are currently in preparation: one is a survey on SSL for image representation learning, with an emphasis on methods adapted for small object detection, and the other investigates the integration of the *a contrario* paradigm and SSL within a YOLO framework, which has achieved excellent performance for frugal settings for IRSTD.

8.2 Perspectives

Based on the literature on deep learning for small target detection and the research conducted during this thesis, we have identified several future directions. We propose to divide these into two categories: first, recommendations for further enhancing the small target detector, and second, perspectives related to defense application needs.

8.2.1 Improving the small object detector...

... using attention mechanisms – To reduce false alarms in challenging environments such as in IRSTD-850 dataset, a comprehensive understanding of the scene is crucial. In the literature, attention mechanisms have been proposed to address this issue. For instance, AGPCNet computes global attention scores at the patch level to highlight the regions that are more likely to contain a target. Considering that ViTs are known for efficiently modelling long-range dependencies, it raises the following question: could the use of a pure ViT encoder be beneficial in this scenario?

Backbone init.	NUAA-SIRST		IRSTD-850	
	F1	**AP**	**F1**	**AP**
YOLO baselines				
YOLOv7-tiny	96.5	97.8	82.2	85.0
YOLO-R50	97.5	98.1	82.3	84.3
Integrating a Swin Transformer into YOLO				
YOLO-Swin	91.4	94.8	81.6	83.0
YOLO-Swin + SSL (ReSim)	94.3	96.0	84.8	87.5
YOLO-Swin+NFA$_{\mathcal{N}}$	97.3	98.0	86.6	88.5

Table 8.3: Performance of YOLO-Swin on SIRST dataset.

To explore this idea, we conducted some tests by replacing the YOLOv7-tiny backbone with the tiny version of the Swin Transformer, leading to a new network that we named YOLO-Swin. The results are shown in Table 8.3. We can see that YOLO-Swin performs significantly worse on the SIRST dataset compared to other YOLO baselines. On the IRSTD-850 dataset, the performance gap between YOLO-Swin and the other YOLO baselines is smaller. The use of SSL pre-trained weights or the NFA detection head notably improves performance, bringing the results closer to the best YOLO baselines (i.e., YOLO + NFA).

This suggests that using a Transformer-based backbone "as is" may not be optimal for IRSTD. It is likely that the architecture needs adaptation, especially to minimise information loss for small targets. One possible approach is to combine convolutional layers and ViT, using convolutions to extract fine-scale local information and ViT to model long-range dependencies at a coarser scale. MTU-Net, for example, adopts this strategy, but as shown in Table 8.1, it performs poorly on the SIRST dataset. Several factors may contribute to these results. First, MTU-Net was originally designed for large input images (1024×1024), and we scaled it down to 256×256 without adjusting the patch size in the ViT blocks. It is possible that using smaller patch sizes could enhance performance. Second, MTU-Net relies on max pooling for each downsampling step, unlike other architectures (e.g., ResNet encoder in DNANet) that use a convolutional layer with a stride of 2 instead. This reliance on max pooling is likely to result in information loss for small objects. Therefore, there is potential for improvement in leveraging ViT for infrared small target detection.

... by relying on knowledge transfer – In Part II of the manuscript, we have shown that using SSL pre-trained weights for the encoder improves performance. However, the pretext tasks and fine-tuning strategies currently in use may be sub-optimal. To better suit our needs for detecting rare and small objects in infrared

images, the pretext tasks should be adapted accordingly. For instance, to make the SSL method more robust to frugal setting *and* to pre-training on a custom in-domain dataset, we should consider combining several SSL approaches, such as MIM methods with local or global instance discrimination methods. Additionally, a pretext task focused on anomaly detection and background estimation, as suggested in [218], could be beneficial for our task. Moreover, if transformer-based detectors prove to be effective for IRSTD, it would be valuable to analyse the attention maps and perform MIM pre-training with a carefully designed masking strategy.

Another important consideration is which part of the detector should be pre-trained. Typically, in the literature, it is common to train the encoder only, with the aim of having a versatile encoder suitable for various tasks, including classification. However, it may be interesting to also pre-train part of the decoder, either by incorporating part of the reconstruction head for MIM methods or by designing a pretext task that involves both an encoder and a decoder. This is, for example, what is proposed in DETReg [219], where the entire detection network (DETR) is trained using the SSL paradigm. In this case, the pretext task involves proposing regions of interest (i.e., areas that are likely to contain an object), with the ground truth generated by unsupervised methods such as selective search.

Furthermore, we have seen that the fine-tuning strategy is very important in order to correctly transfer the knowledge. One straightforward yet effective approach is to freeze the weights of the pre-trained encoder during fine-tuning on the downstream task. However, this strategy is only efficient if the rest of the network (e.g., the decoder or detection head) has sufficient layers, and it may not be the most optimal approach. This raises questions about which layers should be fine-tuned, at what point in the training process, and whether the fine-tuning strategy should be adjusted according to the pre-training method. Some strategies suggested in the literature include multi-step fine-tuning, using different learning rates for each network component (as done in [220]), or regularising the pre-trained weights during fine-tuning.

Finally, an interesting research area consists in leveraging the knowledge from foundation models, such as GPT-3 [221] or Segment Anything Model (SAM [222]). These models have been trained on vast textual or image datasets, often using self-supervised learning techniques. Some initial attempts have been made to apply SAM to infrared small target segmentation [223, 224], revealing several challenges associated with using such models. For example, how can an appropriate tuning prompt be selected? Additionally, how can the knowledge from a large foundation model be effectively distilled into a smaller model without causing overfitting to the IRSTD task? Nevertheless, the results presented in [223, 224] are promising and suggest potential for leveraging foundation models to enhance small object

detection.

8.2.2 Perspectives linked to defense application requirements

Multi-spectral or temporal detection – In defense applications, the availability of time series and hyperspectral images offers a rich amount of information that can significantly enhance small target detection capabilities. These data types provide advantages such as increased spectral information and the ability to observe targets over multiple frames, which can help temporal filtering of false alarms. To fully exploit this potential, it would be interesting to adapt our detection models to better utilise these data. A straightforward approach is to stack the time series or spectral channels depth-wise, then modify the first layers of the encoder to process this information in-depth rather than spatially. This can be accomplished with 1D convolutions over the temporal or spectral dimensions [225, 226]. Attention mechanisms can also be integrated in order to capture dependencies more efficiently, as done with SpectralFormer [227] backbone. This strategy allows the rest of the detector to remain unchanged.

In the case where the temporal filtering is applied *after* performing single-frame detection, it may be prudent to avoid using the NFA detection head to perform the single-frame detection. Indeed, the aim should be to maximise detections, accepting an increase in false alarms, which will be subsequently reduced through temporal filtering. The NFA approach, which prioritises precision at the cost of potential missed detection, may not be optimal in such scenarios.

Furthermore, the integration of temporal and spectral information also opens up new opportunities for designing pretext tasks that leverage these rich data sources. For instance, pretext tasks could involve predicting the next frame in a sequence or ensuring invariance to the dropping or permutation of spectral bands [228]. This promising research direction will be further explored by Ambroise Bouru−−Gazeau in his thesis.

Towards a versatile target detector – While it is crucial to detect targets as early as possible, this does not mean we should overlook detecting closer, and *a fortiori* large targets. One issue with our NFA-based detector is that it excels at detecting very small targets but becomes less effective when the targets occupy a large portion of the image. To address this limitation, it would be valuable to combine multiple detectors, each optimised for different target sizes. For instance, we could use a baseline detector (e.g., YOLOv7-tiny) that performs well on large objects alongside our YOLOv7-tiny + NFA detector, which specialises in small object detection.

One approach could be to explore ensemble methods to merge the detections from these different detectors. However, given that the detectors have distinct

roles, traditional ensemble strategies such as majority voting or weighted voting might not be appropriate. Instead, a more tailored approach, such as confidence-based combination of elementary detections, could be developed to balance the strengths of each detector. Another possibility is to cascade the detectors (as done with Cascade R-CNN [229]), where one detector processes the image first and passes on the remaining task to the next detector based on certain criteria, such as target size.

Combining multiple detectors not only improves detection accuracy across various target sizes but also provides additional information that can be used to explain the detection process. This is an important step towards improving explainability and uncertainty measurement, which are critical aspects for our application, particularly in defense settings where understanding and justifying detections are as important as the detections themselves. Additionally, this approach could facilitate more robust performance in complex and diverse environments, where the range of target sizes and conditions can vary widely.

Appendices

Appendix A

CKA maps

In the literature, the Centered Kernel Alignment (CKA) is a metric that is commonly used in order to analyse the layer-wise similarity between the features extracted by two models. Our approach is based on [202]: to compute the similarity between features across several batches (since the computation across an entire dataset is too computationally expensive) we use an unbiased estimator of the Hilbert-Schmidt Independence Criterion (HSIC [230]) provided by [231]. Given a mini-batch j of n samples, the HSIC between two flattened features $\mathbf{X_j}$ and $\mathbf{Y_j}$ (each with dimension $(n, C \times H \times W)$, with C the number of channels, H the height and W the width) is computed as follows:

$$\mathrm{HSIC}_j(\mathbf{K_j}, \mathbf{L_j}) = \frac{1}{n(n-3)} \left(\mathrm{tr}(\tilde{\mathbf{K}}_\mathbf{j}\tilde{\mathbf{L}}_\mathbf{j}) + \frac{\mathbb{1}^\intercal \tilde{\mathbf{K}}_\mathbf{j} \mathbb{1} \mathbb{1}^\intercal \tilde{\mathbf{L}}_\mathbf{j} \mathbb{1}}{(n-1)(n-2)} - \frac{2}{n-2} \mathbb{1}^\intercal \tilde{\mathbf{K}}_\mathbf{j} \tilde{\mathbf{L}}_\mathbf{j} \mathbb{1} \right),$$

where $\mathbf{K_j} = \mathbf{X}\mathbf{X}^T$ and $\mathbf{L_j} = \mathbf{Y_j}\mathbf{Y_j}^T$, $\mathbb{1}$ is the identity matrix, $\mathrm{tr}(\cdot)$ is the trace of a matrix, and where $\tilde{\mathbf{K}}_\mathbf{j}$ and $\tilde{\mathbf{L}}_\mathbf{j}$ are the respective matrices $\mathbf{K_j}$ and $\mathbf{L_j}$ with the diagonal numbers replaced by 0. An HSIC close to 0 indicates the independence of the observations. This similarity is independent of the permutation of vectors in the representations, which is important when considering feature maps with several channels. The normalised similarity CKA averaged over all the mini-batches (k mini-batches, [215]) is therefore defined as:

$$\mathrm{CKA} = \frac{\frac{1}{k}\sum_{j=1}^{k} \mathrm{HSIC}_j(\mathbf{K_j}, \mathbf{L_j})}{\sqrt{\frac{1}{k}\sum_{j=1}^{k} \mathrm{HSIC}_j(\mathbf{K_j}, \mathbf{K_j})} \sqrt{\frac{1}{k}\sum_{j=1}^{k} \mathrm{HSIC}_j(\mathbf{L_j}, \mathbf{L_j})}}$$

In practice, we take the last layer of each convolution block of the ResNet-50 backbone, giving a total of 18 layers. In our experiments, we visualise the layer-wise similarity between two models using heatmaps. We refer to "self-similarity" when

computing the similarity between feature maps extracted by the *same* network. The diagonal of the CKA map is then necessarily equal to 1. The layers are ranked from 0 to 18, and ordered from the lowest level (close to the input) to the highest level (more abstract layers).

Appendix B

Publications

Papers

PR-2024 Alina Ciocarlan, Sylvie Le Hegarat-Mascle, Sidonie Lefebvre, and Arnaud Woiselle. Deep-NFA: A deep a contrario framework for tiny object detection. Pattern Recognition, 150:110312, 2024

under review Alina Ciocarlan, Sylvie Le Hegarat-Mascle, Sidonie Lefebvre, and Arnaud Woiselle. Robust infrared small target detection using self-supervised and *a contrario* paradigms

under review Alina Ciocarlan, Sidonie Lefebvre, Sylvie Le Hegarat-Mascle, and Arnaud Woiselle. Self-Supervised Learning for Real-World Object Detection: a Survey.

Conferences

GRETSI-2022 Alina Ciocarlan, Sylvie Le Hégarat-Mascle, Sidonie Lefebvre, and Clara Barbanson. Détection de petites cibles par apprentissage profond et critère a contrario. In GRETSI 2022, 2022

ICASSP-2024 Alina Ciocarlan, Sylvie Le Hegarat-Mascle, Sidonie Lefebvre, Arnaud Woiselle, and Clara Barbanson. A contrario paradigm for yolo-based infrared small target detection. In ICASSP 2024 - IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5630–5634. IEEE, 2024

Best PhD student paper award - OPTRO 2024

OPTRO-2024 Alina Ciocarlan, Sylvie Le Hegarat-Mascle, Sidonie Lefebvre, Arnaud Woiselle, and Clara Barbanson. A contrario paradigm for infrared small target detection. In OPTRO 2024, 2024

Bibliography

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In International Conference on Learning Representations, 2021.

[2] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF international conference on computer vision, pages 10012–10022, 2021.

[3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28:91–99, 2015.

[4] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pages 234–241. Springer, 2015.

[6] Yimian Dai, Yiquan Wu, Fei Zhou, and Kobus Barnard. Asymmetric contextual modulation for infrared small target detection. In Proceedings of the IEEE/CVF winter conference on applications of computer vision, pages 950–959, 2021.

[7] Mingjin Zhang, Rui Zhang, Yuxiang Yang, Haichen Bai, Jing Zhang, and Jie Guo. Isnet: Shape matters for infrared small target detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 877–886, 2022.

[8] Boyang Li, Chao Xiao, Longguang Wang, Yingqian Wang, Zaiping Lin, Miao Li, Wei An, and Yulan Guo. Dense nested attention network for infrared small target detection. IEEE Transactions on Image Processing, 2022.

[9] Huan Wang, Luping Zhou, and Lei Wang. Miss detection vs. false alarm: Adversarial learning for small object segmentation in infrared images. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 8509–8518, 2019.

[10] Sebastien Razakarivony and Frederic Jurie. Vehicle detection in aerial imagery: A small target detection benchmark. Journal of Visual Communication and Image Representation, 34:187–203, 2016.

[11] Alina Ciocarlan and Andrei Stoian. Ship Detection in Sentinel 2 Multi-Spectral Images with Self-Supervised Learning. Remote Sensing, 13(21):4255, October 2021.

[12] Agnes Desolneux, Lionel Moisan, and Jean-Michel Morel. From gestalt theory to image analysis: a probabilistic approach, volume 34. Springer Science & Business Media, 2007.

[13] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 11531–11539, 2020.

[14] Niki Parmar, Prajit Ramachandran, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. In Advances in Neural Information Processing Systems, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pages 68–80, 2019.

[15] Renke Kou, Chunping Wang, Zhenming Peng, Zhihe Zhao, Yaohong Chen, Jinhui Han, Fuyu Huang, Ying Yu, and Qiang Fu. Infrared small target segmentation networks: A survey. Pattern Recognition, 143:109788, 2023.

[16] Xiaoping Wang and Tianxu Zhang. Clutter-adaptive infrared small target detection in infrared maritime scenarios. Optical Engineering, 50(6):067001–067001, 2011.

[17] Lizhen Deng, Hu Zhu, Chao Tao, and Yantao Wei. Infrared moving point target detection based on spatial–temporal local contrast filter. Infrared Physics & Technology, 76:168–173, 2016.

[18] Victor T Tom, Tamar Peli, May Leung, and Joseph E Bondaryk. Morphology-based algorithm for point target detection in infrared backgrounds. In Signal and Data Processing of Small Targets 1993, volume 1954, pages 2–11. SPIE, 1993.

[19] Suyog D Deshpande, Meng Hwa Er, Ronda Venkateswarlu, and Philip Chan. Max-mean and max-median filters for detection of small targets. In Signal and Data Processing of Small Targets 1999, volume 3809, pages 74–83. SPIE, 1999.

[20] Xiaoyang Wang, Zhenming Peng, Ping Zhang, and Yanmin He. Infrared small target detection via nonnegativity-constrained variational mode decomposition. IEEE geoscience and remote sensing letters, 14(10):1700–1704, 2017.

[21] CL Philip Chen, Hong Li, Yantao Wei, Tian Xia, and Yuan Yan Tang. A local contrast method for small infrared target detection. IEEE transactions on geoscience and remote sensing, 52(1):574–581, 2013.

[22] Ke Shang, Xiao Sun, Jinwen Tian, Yansheng Li, and Jiayi Ma. Infrared small target detection via line-based reconstruction and entropy-induced suppression. Infrared Physics & Technology, 76:75–81, 2016.

[23] Yao Qin, Lorenzo Bruzzone, Chengqiang Gao, and Biao Li. Infrared small target detection based on facet kernel and random walker. IEEE Transactions on Geoscience and Remote Sensing, 57(9):7104–7118, 2019.

[24] Sur Singh Rawat, Sashi Kant Verma, and Yatindra Kumar. Review on recent development in infrared small target detection algorithms. Procedia Computer Science, 167:2496–2505, 2020.

[25] Mingjing Zhao, Wei Li, Lu Li, Jin Hu, Pengge Ma, and Ran Tao. Single-frame infrared small-target detection: A survey. IEEE Geoscience and Remote Sensing Magazine, 10(2):87–119, 2022.

[26] Lian Huang, Shaosheng Dai, Tao Huang, Xiangkang Huang, and Haining Wang. Infrared small target segmentation with multiscale feature representation. Infrared Physics & Technology, 116:103755, 2021.

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision, pages 2980–2988, 2017.

[28] Luc Courtrai, Minh-Tan Pham, and Sébastien Lefèvre. Small object detection in remote sensing images based on super-resolution with auxiliary generative adversarial networks. Remote Sensing, 12(19):3152, 2020.

[29] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In Jennifer Dy and Andreas Krause, editors, Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, pages 4393–4402. PMLR, 10–15 Jul 2018.

[30] Agnès Desolneux, Lionel Moisan, and J-M Morel. A grouping principle and four applications. IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(4):508–513, 2003.

[31] Thibaud Ehret, Axel Davy, Mauricio Delbracio, and Jean-Michel Morel. How to reduce anomaly detection in images to anomaly detection in noise. Image Processing On Line, 9:391–412, 2019.

[32] Xingang Mou, Shuai Lei, and Xiao Zhou. Yolo-fr: A yolov5 infrared small target detection algorithm based on feature reassembly sampling method. Sensors, 23(5):2710, 2023.

[33] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

[34] Adrian Ziegler and Yuki M Asano. Self-supervised learning of object parts for semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14502–14511, 2022.

[35] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pages 740–755. Springer, 2014.

[36] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 2012.

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[38] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4700–4708, 2017.

[39] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning, pages 6105–6114. PMLR, 2019.

[40] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 11976–11986, 2022.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30:6000–6010, 2017.

[42] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. ArXiv preprint, abs/2102.04306, 2021.

[43] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 580–587, 2014.

[44] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. International journal of computer vision, 104:154–171, 2013.

[45] Ross Girshick. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 1440–1448, 2015.

[46] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2117–2125, 2017.

[47] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference,

Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, pages 21–37. Springer, 2016.

[48] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.

[49] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696, 2022.

[50] Dillon Reis, Jordan Kupec, Jacqueline Hong, and Ahmad Daoudi. Real-time flying object detection with yolov8. arXiv preprint arXiv:2305.09972, 2023.

[51] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. Yolov9: Learning what you want to learn using programmable gradient information. arXiv preprint arXiv:2402.13616, 2024.

[52] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In European conference on computer vision, pages 213–229. Springer, 2020.

[53] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16965–16974, 2024.

[54] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3431–3440, 2015.

[55] Tongle Fan, Guanglei Wang, Yan Li, and Hongrui Wang. MA-Net: A Multi-Scale Attention Network for Liver and Tumor Segmentation. IEEE Access, 8:179656–179665, 2020.

[56] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence, 40(4):834–848, 2017.

[57] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969, 2017.

[58] Md Atiqur Rahman and Yang Wang. Optimizing intersection-over-union in deep neural networks for image segmentation. In International symposium on visual computing, pages 234–244. Springer, 2016.

[59] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[60] Wei Zhang, Mingyu Cong, and Liping Wang. Algorithms for optical weak small targets detection and tracking: review. In International Conference on Neural Networks and Signal Processing, 2003. Proceedings of the 2003, volume 1, pages 643–647 Vol.1, 2003.

[61] Yongbo Cheng, Xuefeng Lai, Yucheng Xia, and Jinmei Zhou. Infrared dim small target detection networks: A review. Sensors, 24(12):3885, 2024.

[62] Xin Wu, Danfeng Hong, and Jocelyn Chanussot. Uiu-net: U-net in u-net for infrared small object detection. IEEE Transactions on Image Processing, 32:364–376, 2022.

[63] Chun Bao, Jie Cao, Yaqian Ning, Tianhua Zhao, Zhijun Li, Zechen Wang, Li Zhang, and Qun Hao. Improved dense nested attention network based on transformer for infrared small target detection. arXiv preprint arXiv:2311.08747, 2023.

[64] Tianfang Zhang, Lei Li, Siying Cao, Tian Pu, and Zhenming Peng. Attention-guided pyramid context networks for detecting infrared small target under complex background. IEEE Transactions on Aerospace and Electronic Systems, 59(4):4250–4261, 2023.

[65] Tianhao Wu, Boyang Li, Yihang Luo, Yingqian Wang, Chao Xiao, Ting Liu, Jungang Yang, Wei An, and Yulan Guo. Mtu-net: Multilevel transunet for space-based infrared tiny ship detection. IEEE Transactions on Geoscience and Remote Sensing, 61:1–15, 2023.

[66] Gong Cheng, Xiang Yuan, Xiwen Yao, Kebing Yan, Qinghua Zeng, Xingxing Xie, and Junwei Han. Towards large-scale small object detection: Survey and benchmarks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023.

[67] Ronghao Li and Ying Shen. Yolosr-ist: A deep learning method for small target detection in infrared remote sensing images based on super-resolution and yolo. Signal Processing, 208:108962, 2023.

[68] Bo Yang, Xinyu Zhang, Jian Zhang, Jun Luo, Mingliang Zhou, and Yangjun Pi. Eflnet: Enhancing feature learning network for infrared small target detection. IEEE Transactions on Geoscience and Remote Sensing, 62:1–11, 2024.

[69] H. von Helmholtz. Treatise on Physiological Optics. 1999.

[70] K. Koffka. Principles of Gestalt psychology. 1923.

[71] M. Wertheimer. Untersuchungen zur lehre der gestalt, II. 1923.

[72] Bénédicte Grosjean and Lionel Moisan. A-contrario Detectability of Spots in Textured Backgrounds. Journal of Mathematical Imaging and Vision, 33(3):313–337, March 2009.

[73] Charles W Dunnett and Ajit C Tamhane. Power comparisons of some step-up multiple test procedures. Statistics & probability letters, 16(1):55–58, 1993.

[74] Joseph P Romano and Michael Wolf. Exact and approximate stepdown methods for multiple hypothesis testing. Journal of the American Statistical Association, 100(469):94–108, 2005.

[75] Sylvie Le Hégarat-Mascle, Emanuel Aldea, and Jennifer Vandoni. Efficient evaluation of the number of false alarm criterion. EURASIP J. Image Video Process., 2019:35, 2019.

[76] Alireza Rezaei, Sylvie Le Hégarat-Mascle, Emanuel Aldea, Piercarlo Dondi, and Marco Malagodi. A-contrario framework for detection of alterations in varnished surfaces. J. Vis. Commun. Image Represent., 83:103357, 2022.

[77] Vincent Vidal, Matthieu Limbert, Tugdual Ceillier, and Lionel Moisan. Aggregated primary detectors for generic change detection in satellite images. In IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium, pages 59–62. IEEE, 2019.

[78] Milton Abramowitz. Handbook of Mathematical Functions, With Formulas, Graphs, and Mathematical Tables,. Dover Publications, Inc., USA, 1974.

[79] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In Artificial intelligence and statistics, pages 562–570. Pmlr, 2015.

[80] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, pages 764–773. IEEE Computer Society, 2017.

[81] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pages 7794–7803. IEEE Computer Society, 2018.

[82] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[83] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7132–7141, 2018.

[84] Mingjin Zhang, Rui Zhang, Jing Zhang, Jie Guo, Yunsong Li, and Xinbo Gao. Dim2clear network for infrared small target detection. IEEE Transactions on Geoscience and Remote Sensing, 61:1–14, 2023.

[85] Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. Road extraction by deep residual u-net. IEEE Geoscience and Remote Sensing Letters, 15(5):749–753, 2018.

[86] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, volume 70 of Proceedings of Machine Learning Research, pages 1321–1330. PMLR, 2017.

[87] Jacob König, Mark David Jenkins, Peter Barrie, Mike Mannion, and Gordon Morison. A convolutional neural network for pavement surface crack segmentation using residual connections and attention gating. In 2019 IEEE international conference on image processing (ICIP), pages 1460–1464. IEEE, 2019.

[88] Haifeng Li, Jianping Zong, Jingjing Nie, Zhilong Wu, and Hongyang Han. Pavement crack detection algorithm based on densely connected and deeply supervised network. IEEE Access, 9:11835–11842, 2021.

[89] Rodrigo Rill-García, Eva Dokládalová, and Petr Dokládal. Pixel-accurate road crack detection in presence of inaccurate annotations. Neurocomputing, 480:1–13, 2022.

[90] Zhenjie Liu, Jianming Xu, Jun Li, Antonio Plaza, Shaoquan Zhang, and Lizhe Wang. Moving ship optimal association for maritime surveillance: Fusing ais and sentinel-2 data. IEEE Transactions on Geoscience and Remote Sensing, 60:1–18, 2022.

[91] Qin Zou, Yu Cao, Qingquan Li, Qingzhou Mao, and Song Wang. Cracktree: Automatic crack detection from pavement images. Pattern Recognition Letters, 33(3):227–238, 2012.

[92] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. You only learn one representation: Unified network for multiple tasks. arXiv preprint arXiv:2105.04206, 2021.

[93] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[94] Alina Ciocarlan, Sylvie Le Hegarat-Mascle, Sidonie Lefebvre, and Arnaud Woiselle. Deep-nfa: A deep a contrario framework for tiny object detection. Pattern Recognition, 150:110312, 2024.

[95] Minh-Tan Pham, Luc Courtrai, Chloé Friguet, Sébastien Lefèvre, and Alexandre Baussard. YOLO-Fine: One-Stage Detector of Small Objects Under Various Backgrounds in Remote Sensing Images. page 26, 2020.

[96] Chang Xu, Jinwang Wang, Wen Yang, Huai Yu, Lei Yu, and Gui-Song Xia. Rfla: Gaussian receptive field based label assignment for tiny object detection. In European conference on computer vision, pages 526–543. Springer, 2022.

[97] Xue Yang, Yue Zhou, Gefan Zhang, Jirui Yang, Wentao Wang, Junchi Yan, XIAOPENG ZHANG, and Qi Tian. The kfiou loss for rotated object detection. In The Eleventh International Conference on Learning Representations.

[98] Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A cookbook of self-supervised learning. arXiv preprint arXiv:2304.12210, 2023.

[99] Tobias Uelwer, Jan Robine, Stefan Sylvius Wagner, Marc Höftmann, Eric Upschulte, Sebastian Konietzny, Maike Behrendt, and Stefan Harmeling. A survey on self-supervised representation learning. arXiv preprint arXiv:2308.11455, 2023.

[100] Utku Ozbulak, Hyun Jung Lee, Beril Boga, Esla Timothy Anzaku, Homin Park, Arnout Van Messem, Wesley De Neve, and Joris Vankerschaver. Know your self-supervised learning: A survey on image-based generative and discriminative training. arXiv preprint arXiv:2305.13689, 2023.

[101] Jie Gui, Tuo Chen, Jing Zhang, Qiong Cao, Zhenan Sun, Hao Luo, and Dacheng Tao. A survey on self-supervised learning: Algorithms, applications, and future trends. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024.

[102] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552, 2017.

[103] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the IEEE/CVF international conference on computer vision, pages 6023–6032, 2019.

[104] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412, 2017.

[105] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 9729–9738, 2020.

[106] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In International conference on machine learning, pages 1597–1607. PMLR, 2020.

[107] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 6707–6717, 2020.

[108] Jovana Mitrovic, Brian McWilliams, Jacob C Walker, Lars Holger Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms. In International Conference on Learning Representations, 2020.

[109] Nenad Tomasev, Ioana Bica, Brian McWilliams, Lars Holger Buesing, Razvan Pascanu, Charles Blundell, and Jovana Mitrovic. Pushing the limits of

self-supervised resnets: Can we outperform supervised learning without labels on imagenet? In First Workshop on Pre-training: Perspectives, Pitfalls, and Paths Forward at ICML 2022, 2022.

[110] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. Advances in neural information processing systems, 33:11839–11852, 2020.

[111] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In European Conference on Computer Vision, pages 392–408. Springer, 2022.

[112] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 113–123, 2019.

[113] Colorado J Reed, Sean Metzger, Aravind Srinivas, Trevor Darrell, and Kurt Keutzer. Selfaugment: Automatic augmentation policies for self-supervised learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 2674–2683, 2021.

[114] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), volume 1, pages 539–546. IEEE, 2005.

[115] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. Journal of machine learning research, 10(2), 2009.

[116] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. Advances in neural information processing systems, 29, 2016.

[117] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748, 2018.

[118] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 815–823, 2015.

[119] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 2495–2504, 2021.

[120] Chaoning Zhang, Kang Zhang, Trung X Pham, Axi Niu, Zhinan Qiao, Chang D Yoo, and In So Kweon. Dual temperature helps contrastive learning without many negative samples: Towards understanding and simplifying moco. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14441–14450, 2022.

[121] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In 2018 IEEE international conference on robotics and automation (ICRA), pages 1134–1141. IEEE, 2018.

[122] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9588–9597, 2021.

[123] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. Advances in neural information processing systems, 33:8765–8775, 2020.

[124] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems, 33:21271–21284, 2020.

[125] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 15750–15758, 2021.

[126] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In Proceedings of the IEEE/CVF international conference on computer vision, pages 9650–9660, 2021.

[127] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Image bert pre-training with online tokenizer. In International Conference on Learning Representations, 2021.

[128] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193, 2023.

[129] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In International Conference on Machine Learning, pages 12310–12320. PMLR, 2021.

[130] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In International Conference on Machine Learning, pages 3015–3024. PMLR, 2021.

[131] Adrien Bardes, Jean Ponce, and Yann Lecun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In ICLR 2022-International Conference on Learning Representations, 2022.

[132] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In Proceedings of the European conference on computer vision (ECCV), pages 132–149, 2018.

[133] YM Asano, C Rupprecht, and A Vedaldi. Self-labelling via simultaneous clustering and representation learning. In International Conference on Learning Representations, 2019.

[134] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In European conference on computer vision, pages 268–285. Springer, 2020.

[135] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. Advances in neural information processing systems, 33:9912–9924, 2020.

[136] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Mike Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. In European Conference on Computer Vision, pages 456–473. Springer, 2022.

[137] Senthil Purushwalkam and Abhinav Gupta. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. Advances in Neural Information Processing Systems, 33:3407–3418, 2020.

[138] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In Proceedings of the IEEE international conference on computer vision, pages 1422–1430, 2015.

[139] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In European conference on computer vision, pages 69–84. Springer, 2016.

[140] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In ICLR 2018, 2018.

[141] Zhenda Xie, Zigang Geng, Jingcheng Hu, Zheng Zhang, Han Hu, and Yue Cao. Revealing the dark secrets of masked image modeling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14475–14485, 2023.

[142] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning, pages 1096–1103, 2008.

[143] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2536–2544, 2016.

[144] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14, pages 649–666. Springer, 2016.

[145] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 6874–6883, 2017.

[146] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. In International Conference on Learning Representations, 2021.

[147] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 16000–16009, 2022.

[148] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 15619–15629, 2023.

[149] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9653–9663, 2022.

[150] Zhaoxu Li, Yingqian Wang, Chao Xiao, Qiang Ling, Zaiping Lin, and Wei An. You only train once: Learning a general anomaly enhancement network with random masks for hyperspectral anomaly detection. IEEE Transactions on Geoscience and Remote Sensing, 61:1–18, 2023.

[151] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 8769–8778, 2018.

[152] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. International Journal of Computer Vision, 127:302–321, 2019.

[153] Siyuan Li, Di Wu, Fang Wu, Zelin Zang, and Stan Z Li. Architecture-agnostic masked image modeling from vit back to cnn. In International Conference on Machine Learning, pages 20149–20167. PMLR, 2023.

[154] Keyu Tian, Yi Jiang, Chen Lin, Liwei Wang, Zehuan Yuan, et al. Designing bert for convolutional networks: Sparse and hierarchical masked modeling. In The Eleventh International Conference on Learning Representations, 2022.

[155] Peng Gao, Teli Ma, Hongsheng Li, Ziyi Lin, Jifeng Dai, and Yu Qiao. Convmae: Masked convolution meets masked autoencoders. arXiv preprint arXiv:2205.03892, 2022.

[156] Jihao Liu, Xin Huang, Jinliang Zheng, Yu Liu, and Hongsheng Li. Mixmae: Mixed and masked autoencoder for efficient pretraining of hierarchical vision transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6252–6261, 2023.

[157] Yuxin Fang, Li Dong, Hangbo Bao, Xinggang Wang, and Furu Wei. Corrupted image modeling for self-supervised visual pre-training. In The Eleventh International Conference on Learning Representations, 2022.

[158] Yuan Liu, Songyang Zhang, Jiacheng Chen, Kai Chen, and Dahua Lin. Pixmim: Rethinking pixel reconstruction in masked image modeling. Transactions on Machine Learning Research, 2024.

[159] Zhaowen Li, Zhiyang Chen, Fan Yang, Wei Li, Yousong Zhu, Chaoyang Zhao, Rui Deng, Liwei Wu, Rui Zhao, Ming Tang, et al. Mst: Masked self-supervised transformer for visual representation. Advances in Neural Information Processing Systems, 34:13165–13176, 2021.

[160] Ioannis Kakogeorgiou, Spyros Gidaris, Bill Psomas, Yannis Avrithis, Andrei Bursuc, Konstantinos Karantzalos, and Nikos Komodakis. What to hide from your students: Attention-guided masked image modeling. In European Conference on Computer Vision, pages 300–318. Springer, 2022.

[161] Zhengqi Liu, Jie Gui, and Hao Luo. Good helper is around you: Attention-driven masked image modeling. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, pages 1799–1807, 2023.

[162] Zejiang Hou, Fei Sun, Yen-Kuang Chen, Yuan Xie, and Sun-Yuan Kung. Milan: Masked image pretraining on language assisted representation. arXiv preprint arXiv:2208.06049, 2022.

[163] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In International conference on machine learning, pages 8748–8763. PMLR, 2021.

[164] Gang Li, Heliang Zheng, Daqing Liu, Chaoyue Wang, Bing Su, and Changwen Zheng. Semmae: Semantic-guided masking for learning masked autoencoders. Advances in Neural Information Processing Systems, 35:14290–14302, 2022.

[165] Junde Xu, Zikai Lin, Donghao Zhou, Yaodong Yang, Xiangyun Liao, Qiong Wang, Bian Wu, Guangyong Chen, and Pheng-Ann Heng. Dppmask: Masked image modeling with determinantal point processes. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 2266–2276, 2024.

[166] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In International Conference on Machine Learning, pages 8821–8831. PMLR, 2021.

[167] Hao Liu, Xinghua Jiang, Xin Li, Antai Guo, Yiqing Hu, Deqiang Jiang, and Bo Ren. The devil is in the frequency: Geminated gestalt autoencoder for self-supervised visual pre-training. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, pages 1649–1656, 2023.

[168] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14668–14678, 2022.

[169] Chaoqin Huang, Qinwei Xu, Yanfeng Wang, Yu Wang, and Ya Zhang. Self-supervised masking for unsupervised anomaly detection and localization. IEEE Transactions on Multimedia, 2022.

[170] Zhiliang Peng, Li Dong, Hangbo Bao, Furu Wei, and Qixiang Ye. A unified view of masked image modeling. Transactions on Machine Learning Research, 2022.

[171] Hongwei Xue, Peng Gao, Hongyang Li, Yu Qiao, Hao Sun, Houqiang Li, and Jiebo Luo. Stare at what you see: Masked image modeling without reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 22732–22741, 2023.

[172] Alaaeldin El-Nouby, Gautier Izacard, Hugo Touvron, Ivan Laptev, Hervé Jegou, and Edouard Grave. Are large-scale datasets necessary for self-supervised pre-training? arXiv preprint arXiv:2112.10740, 2021.

[173] Xingbin Liu, Jinghao Zhou, Tao Kong, Xianming Lin, and Rongrong Ji. Exploring target representations for masked autoencoders. arXiv preprint arXiv:2209.03917, 2022.

[174] Zhicheng Huang, Xiaojie Jin, Chengze Lu, Qibin Hou, Ming-Ming Cheng, Dongmei Fu, Xiaohui Shen, and Jiashi Feng. Contrastive masked autoencoders are stronger vision learners. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023.

[175] Byungseok Roh, Wuhyun Shin, Ildoo Kim, and Sungwoong Kim. Spatially consistent representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1144–1153, 2021.

[176] Tete Xiao, Colorado J Reed, Xiaolong Wang, Kurt Keutzer, and Trevor Darrell. Region similarity representation learning. In Proceedings of the

IEEE/CVF International Conference on Computer Vision, pages 10539–10548, 2021.

[177] Yucheng Zhao, Guangting Wang, Chong Luo, Wenjun Zeng, and Zheng-Jun Zha. Self-supervised visual representations learning by contrastive mask prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10160–10169, 2021.

[178] Fangyun Wei, Yue Gao, Zhirong Wu, Han Hu, and Stephen Lin. Aligning pretraining for detection via object-level contrastive learning. Advances in Neural Information Processing Systems, 34:22682–22694, 2021.

[179] Ramprasaath R Selvaraju, Karan Desai, Justin Johnson, and Nikhil Naik. Casting your model: Learning to localize improves self-supervised representations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11058–11067, 2021.

[180] Xiangyu Peng, Kai Wang, Zheng Zhu, Mang Wang, and Yang You. Crafting better contrastive views for siamese representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16031–16040, 2022.

[181] Ceyuan Yang, Zhirong Wu, Bolei Zhou, and Stephen Lin. Instance localization for self-supervised detection pretraining. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3987–3996, 2021.

[182] Feng Wang, Huiyu Wang, Chen Wei, Alan Yuille, and Wei Shen. Cp 2: Copy-paste contrastive pretraining for semantic segmentation. In European Conference on Computer Vision, pages 499–515. Springer, 2022.

[183] Jiahao Xie, Xiaohang Zhan, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Unsupervised object-level representation learning from scene images. Advances in Neural Information Processing Systems, 34:28864–28876, 2021.

[184] Junwei Yang, Ke Zhang, Zhaolin Cui, Jinming Su, Junfeng Luo, and Xiaolin Wei. Inscon: Instance consistency feature representation via self-supervised learning. arXiv preprint arXiv:2203.07688, 2022.

[185] Pedro O O Pinheiro, Amjad Almahairi, Ryan Benmalek, Florian Golemo, and Aaron C Courville. Unsupervised learning of dense visual representations. Advances in Neural Information Processing Systems, 33:4489–4500, 2020.

[186] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16684–16693, 2021.

[187] Jian Ding, Enze Xie, Hang Xu, Chenhan Jiang, Zhenguo Li, Ping Luo, and Gui-Song Xia. Deeply unsupervised patch re-identification for pre-training object detectors. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022.

[188] Ashraful Islam, Benjamin Lundell, Harpreet Sawhney, Sudipta N Sinha, Peter Morales, and Richard J Radke. Self-supervised learning with local contrastive loss for detection and semantic segmentation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 5624–5633, 2023.

[189] Thalles Silva, Helio Pedrini, and Adín Ramírez. Self-supervised learning of contextualized local visual embeddings. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 177–186, 2023.

[190] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3024–3033, 2021.

[191] Songtao Liu, Zeming Li, and Jian Sun. Self-emd: Self-supervised object detection without imagenet. arXiv preprint arXiv:2011.13677, 2020.

[192] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicregl: Self-supervised learning of local visual features. Advances in Neural Information Processing Systems, 35:8799–8810, 2022.

[193] Olivier J Hénaff, Skanda Koppula, Jean-Baptiste Alayrac, Aaron Van den Oord, Oriol Vinyals, and Joao Carreira. Efficient visual pretraining with contrastive detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10086–10096, 2021.

[194] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. International journal of computer vision, 59:167–181, 2004.

[195] Olivier J Hénaff, Skanda Koppula, Evan Shelhamer, Daniel Zoran, Andrew Jaegle, Andrew Zisserman, João Carreira, and Relja Arandjelović. Object discovery and representation networks. In European Conference on Computer Vision, pages 123–143. Springer, 2022.

[196] Zhaoqing Wang, Qiang Li, Guoxin Zhang, Pengfei Wan, Wen Zheng, Nannan Wang, Mingming Gong, and Tongliang Liu. Exploring set similarity for dense self-supervised representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16590–16599, 2022.

[197] Yanghao Li, Saining Xie, Xinlei Chen, Piotr Dollar, Kaiming He, and Ross Girshick. Benchmarking detection transfer learning with vision transformers. arXiv preprint arXiv:2111.11429, 2021.

[198] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In Proceedings of the IEEE/CVF international conference on computer vision, pages 9640–9649, 2021.

[199] Namuk Park, Wonjae Kim, Byeongho Heo, Taekyung Kim, and Sangdoo Yun. What do self-supervised vision transformers learn? In The Eleventh International Conference on Learning Representations, 2022.

[200] Chenxin Tao, Xizhou Zhu, Weijie Su, Gao Huang, Bin Li, Jie Zhou, Yu Qiao, Xiaogang Wang, and Jifeng Dai. Siamese image modeling for self-supervised vision representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2132–2141, 2023.

[201] Dilxat Muhtar, Xueliang Zhang, Pengfeng Xiao, Zhenshi Li, and Feng Gu. Cmid: A unified self-supervised learning framework for remote sensing image understanding. IEEE Transactions on Geoscience and Remote Sensing, 61:1–17, 2023.

[202] Jin Gao, Shubo Lin, Shaoru Wang, Yutong Kou, Zeming Li, Liang Li, Congxuan Zhang, Xiaoqin Zhang, Yizheng Wang, and Weiming Hu. Observation, analysis, and solution: Exploring strong lightweight vision transformers via masked image modeling pre-training. arXiv preprint arXiv:2404.12210, 2024.

[203] Nanxuan Zhao, Zhirong Wu, Rynson WH Lau, and Stephen Lin. What makes instance discrimination good for transfer learning? In International Conference on Learning Representations, 2020.

[204] Sungnyun Kim, Sangmin Bae, and Se-Young Yun. Coreset sampling from open-set for fine-grained self-supervised learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7537–7547, 2023.

[205] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. arXiv:1703.06870, January 2018.

[206] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101, 2017.

[207] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. arXiv preprint arXiv:1706.02677, 2017.

[208] Yuxin Fang, Shusheng Yang, Shijie Wang, Yixiao Ge, Ying Shan, and Xinggang Wang. Unleashing vanilla vision transformer with masked image modeling for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 6244–6253, 2023.

[209] Qiao Liu, Xin Li, Zhenyu He, Chenglong Li, Jun Li, Zikun Zhou, Di Yuan, Jing Li, Kai Yang, Nana Fan, et al. Lsotb-tir: A large-scale high-diversity thermal infrared object tracking benchmark. In Proceedings of the 28th ACM international conference on multimedia, pages 3847–3856, 2020.

[210] Heng Sun, Junxiang Bai, Fan Yang, and Xiangzhi Bai. Receptive-field and direction induced attention network for infrared dim small target detection with a large-scale dataset irdst. IEEE Transactions on Geoscience and Remote Sensing, 61:1–13, 2023.

[211] Thermal Imaging. Flir data set dataset. https://universe.roboflow.com/thermal-imaging-0hwfw/flir-data-set, mar 2024. visited on 2024-07-16.

[212] Jan Portmann, Simon Lynen, Margarita Chli, and Roland Siegwart. People detection and tracking from aerial thermal views. In 2014 IEEE international conference on robotics and automation (ICRA), pages 1794–1800. IEEE, 2014.

[213] Jiashun Suo, Tianyi Wang, Xingzhou Zhang, Haiyang Chen, Wei Zhou, and Weisong Shi. Hit-uav: A high-altitude infrared thermal dataset for unmanned aerial vehicle-based object detection. Scientific Data, 10(1):227, 2023.

[214] Cristina Vasconcelos, Vighnesh Birodkar, and Vincent Dumoulin. Proper reuse of image classification features improves object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 13628–13637, 2022.

[215] Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. In International Conference on Learning Representations, 2021.

[216] Alina Ciocarlan, S Le Hégarat-Mascle, Sidonie Lefebvre, and Clara Barbanson. Détection de petites cibles par apprentissage profond et critère a contrario. In GRETSI 2022, 2022.

[217] Alina Ciocarlan, Sylvie Le Hegarat-Mascle, Sidonie Lefebvre, Arnaud Woiselle, and Clara Barbanson. A contrario paradigm for yolo-based infrared small target detection. In ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 5630–5634. IEEE, 2024.

[218] Neelu Madan, Nicolae-Cătălin Ristea, Radu Tudor Ionescu, Kamal Nasrollahi, Fahad Shahbaz Khan, Thomas B Moeslund, and Mubarak Shah. Self-supervised masked convolutional transformer block for anomaly detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023.

[219] Amir Bar, Xin Wang, Vadim Kantorov, Colorado J Reed, Roei Herzig, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. Detreg: Unsupervised pretraining with region priors for object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 14605–14615, 2022.

[220] Seokhyeon Ha, Sunbeom Jeong, and Jungwoo Lee. Domain-aware fine-tuning: Enhancing neural network adaptability. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, pages 12261–12269, 2024.

[221] Tom B Brown. Language models are few-shot learners. arXiv preprint ArXiv:2005.14165, 2020.

[222] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 4015–4026, 2023.

[223] Mingjin Zhang, Yuchun Wang, Jie Guo, Yunsong Li, Xinbo Gao, and Jing Zhang. Irsam: Advancing segment anything model for infrared small target detection. arXiv preprint arXiv:2407.07520, 2024.

[224] Mingjin Zhang, Chi Zhang, Qiming Zhang, Yunsong Li, Xinbo Gao, and Jing Zhang. Unleashing the power of generic segmentation model: A simple baseline for infrared small target detection. In ACM Multimedia 2024, 2024.

[225] Yushi Chen, Hanlu Jiang, Chunyang Li, Xiuping Jia, and Pedram Ghamisi. Deep feature extraction and classification of hyperspectral images based on

convolutional neural networks. IEEE transactions on geoscience and remote sensing, 54(10):6232–6251, 2016.

[226] Du Tran, Jamie Ray, Zheng Shou, Shih-Fu Chang, and Manohar Paluri. Convnet architecture search for spatiotemporal feature learning. arXiv preprint arXiv:1708.05038, 2017.

[227] Danfeng Hong, Zhu Han, Jing Yao, Lianru Gao, Bing Zhang, Antonio Plaza, and Jocelyn Chanussot. Spectralformer: Rethinking hyperspectral image classification with transformers. IEEE Transactions on Geoscience and Remote Sensing, 60:1–15, 2021.

[228] Nassim Ait Ali Braham, Lichao Mou, Jocelyn Chanussot, Julien Mairal, and Xiao Xiang Zhu. Self supervised learning for few shot hyperspectral image classification. In IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium, pages 267–270. IEEE, 2022.

[229] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 6154–6162, 2018.

[230] Arthur Gretton, Ralf Herbrich, Alexander Smola, Olivier Bousquet, Bernhard Schölkopf, and Aapo Hyvärinen. Kernel methods for measuring independence. Journal of Machine Learning Research, 6(12), 2005.

[231] Le Song, Alex Smola, Arthur Gretton, Justin Bedo, and Karsten Borgwardt. Feature selection via dependence maximization. Journal of Machine Learning Research, 13(47):1393–1434, 2012.