OBJECT DETECTION WITH DEEP NEURAL NETWORKS UNDER CONSTRAINED SCENARIOS

GONGJIE ZHANG

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2022

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

08/08/2022

Date

GONGJIE ZHANG

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

08/08/2022

Date

Assoc. Prof. Shijian Lu

Authorship Attribution Statement

This thesis contains material from three papers published in and two papers submitted to the following peer-reviewed journals / conferences, in which I am listed as the first author for all five papers.

Chapter 3 is published as **Gongjie Zhang**, Shijian Lu, and Wei Zhang. "CAD-Net: A Context-Aware Detection Network for Objects in Remote Sensing Imagery." IEEE Transactions on Geoscience and Remote Sensing (**T-GRS**), vol. 57, no. 12, pp. 10015-10024, 2019.

The contributions of the co-authors are as follows:

- Prof. Shijian Lu pointed out the overall research direction and provided supervision during the research.
- I designed and implemented the algorithm, conducted the experiments, and prepared the manuscript drafts.
- Prof. Shijian Lu and Prof. Wei Zhang provided valuable comments on the manuscript.

Part of Chapter 4 is published as **Gongjie Zhang**, Zhipeng Luo, Yingchen Yu, Kaiwen Cui, and Shijian Lu. "Accelerating DETR Convergence via Semantic-Aligned Matching." in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (**CVPR**). pp. 949-958. 2022.

Part of Chapter 4 is submitted to International Journal of Computer Vision (**IJCV**) for peer review as **Gongjie Zhang**, Zhipeng Luo, Jiaxing Huang, Shijian Lu, and Eric P. Xing. "Semantic-Aligned Matching for Enhanced DETR Convergence and Multi-Scale Feature Fusion."

The contributions of the co-authors are as follows:

- Prof. Shijian Lu pointed out the overall research direction and provided supervision during the research.
- I designed and implemented the algorithm, conducted the experiments, and prepared the manuscript drafts.
- Prof. Shijian Lu, Prof. Eric P. Xing, Zhipeng Luo, Yingchen Yu, Jiaxing Huang, and Kaiwen Cui provided valuable comments on the manuscript.

Chapter 5 is published as **Gongjie Zhang**, Zhipeng Luo, Kaiwen Cui, Shijian Lu, and Eric P. Xing. "Meta-DETR: Image-Level Few-Shot Detection with Inter-Class Correlation Exploitation." IEEE Transactions on Pattern Analysis and Machine Intelligence (**T-PAMI**), 2022.

The contributions of the co-authors are as follows:

- Prof. Shijian Lu pointed out the overall research direction and provided supervision during the research.
- I designed and implemented the algorithm, conducted the experiments, and prepared the manuscript drafts.
- Prof. Shijian Lu, Prof. Eric P. Xing, Zhipeng Luo, and Kaiwen Cui provided valuable comments on the manuscript.

Chapter 6 is submitted to the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2023 (**CVPR 2023**) for peer review as **Gongjie Zhang**, Zhipeng Luo, Zichen Tian, Jingyi Zhang, Xiaoqin Zhang, and Shijian Lu. "Towards Efficient Use of Multi-Scale Features in Transformer-Based Object Detectors."

The contributions of the co-authors are as follows:

- Prof. Shijian Lu pointed out the overall research direction and provided supervision during the research.
- I designed and implemented the algorithm, conducted the experiments, and prepared the manuscript drafts.
- Prof. Shijian Lu, Zhipeng Luo, Zichen Tian, Jingyi Zhang, and Xiaoqin Zhang provided valuable comments on the manuscript.

08/08/2022

Date

GONGJIE ZHANG

Acknowledgements

I wish to express my sincerest gratitude to my supervisor, Prof. Shijian Lu, for his support and patience in guiding me to become a qualified researcher. He set a real model for me with his patience and kindness to students and his devotion to research. During the four years, I have learned a lot and grown so much. Without his guidance and kind advice, I would never be able to go beyond myself and complete this Ph.D. program.

I would like to thank all my collaborators for their valuable suggestions on my research works. I wish to express special thanks to Prof. Lu's team, as well as everyone in the Multimedia and Interactive Computing Lab, Delta-NTU Corp Lab, and S-Lab for Advanced Intelligence. I really enjoy the atmosphere in the team and the labs.

I would like to thank all my friends. Thank you for making my days in Singapore so far a wonderful and memorable time.

I would also like to thank my parents for their endless love for me, as well as their strong support and encouragement as I undertake this Ph.D. program.

I would like to thank Ministry of Education Singapore, Delta Electronics Inc., and SenseTime Group for their financial support in my researches.

Finally, I would like to thank my cat Perper for her companionship and for occasionally moving her butt away from my keyboard so that I could type from time to time.

Gongjie Zhang, August 2022

Contents

List of Figures

List of Tables

Abstract

Object detection, which aims to recognize and locate objects within images using bounding boxes, is one of the most fundamental tasks in computer vision. Object detection forms the basis for many other computer vision tasks and has extensive use cases, such as autonomous driving, surveillance, robotic vision, etc. In the past ten years, object detection has made unprecedented progress with the development of deep neural networks. Compared with prior arts that adopt handcrafted features, modern object detectors benefit from the strong feature representations produced by deep neural networks, and have achieved strong performance on many challenging generic object detection benchmarks, such as MS COCO and Open Images.

However, deep-neural-network-based object detectors are still far from perfect, still facing many challenges under various constrained scenarios. First, modern object detectors heavily rely on visual clues such as texture details, contours, and contrast with the background. However, in some scenarios (e.g., adverse weather or aerial object detection), these features are largely degraded or missing, adding substantial difficulty to object detection. Second, deep-neural-network-based object detectors usually require long training iterations, which are time-consuming and expensive, or even unaffordable to many researchers or companies. Third, as modern object detectors are mostly based on deep neural networks, they require huge amounts of training samples to learn a visual concept. However, such large-scale and annotated datasets are not always available due to expensive human labeling costs or difficulty in data acquisition. Fourth, when deploying modern detectors on edge devices with limited computational capacity, their complexity can be a bottleneck due to runtime requirements.

This thesis focuses on advancing object detection in several constrained scenarios. First, we design a novel Context-Aware Detection Network (CAD-Net) for accurate and robust object detection within optical remote sensing imagery. Generic object detection techniques usually experience a sharp performance drop when directly applied to remote sensing images, largely due to the object appearance differences

in remote sensing images in terms of sparse texture, low contrast, arbitrary orientations, large scale variations, etc. To adapt to this scenario, CAD-Net extracts scene-level and object-level contextual information, which is highly correlated to objects of interest, to provide extra guidance. Besides, a spatial-and-scale-aware attention module is designed to highlight scale-adaptive features and the degraded texture details. Second, we design a novel semantic-aligned matching mechanism to accelerate the convergence of the newly proposed DEtection TRansformer (DETR), which reduces the training iterations by over 95% with improved detection accuracy. Third, we design Meta-DETR for few-shot object detection, which tackles the challenge of training with only a few annotated examples. Meta-DETR fully bypasses the low-quality object proposals for novel classes, thus achieving superior performance to prior R-CNN-based few-shot object detectors. In addition, Meta-DETR performs meta-learning on a set of support classes simultaneously, thus effectively leveraging the inter-class correlation among different classes for better generalization. Fourth, we design a novel paradigm, named Iterative Multi-scale Feature Aggregation (IMFA), to enable the efficient use of multi-scale features in the newly proposed Transformer-based object detectors. Directly incorporating multi-scale features will lead to prohibitive computational costs due to the poor efficiency of the attention mechanism to process high-resolution features. IMFA innovatively exploits sparse multi-scale features only from the most promising and informative locations and significantly improves detection accuracy on multiple object detectors at marginal costs.

Chapter 1

Introduction

1.1 Background

As a longstanding and important task in computer vision, object detection [2–4] has been an active research topic for decades. Its goal is to determine whether there are any object instances from a set of pre-defined categories (like humans, dogs, motorbikes, *etc.*) in an image and, if present, to recognize each object instance and predict the location of each object instance using a bounding box.

As a fundamental task in computer vision, object detection can serve as the cornerstone for many other vision tasks, such as scene understanding, tracking, image captioning, re-identification, *etc.* The use cases involving object detection are also incredibly diverse, with some examples provided below.

- **Autonomous Driving**: Self-driving cars rely heavily on object detection to recognize and locate pedestrians, other vehicles, traffic signals, and road signs. For example, Tesla's Autopilot system involves extensive use of object detection to perceive surrounding threats like oncoming vehicles or obstacles.
- **Security and Surveillance**: Many access control systems and surveillance systems rely on object detection. For example, to detect people in restricted or dangerous areas or to detect human faces for identity verification.
- **Retail**: Object detection can be used in unmanned stores to analyze customers' behavior and to perform automatic checkout without any cashier.

- **Healthcare**: Medical diagnostics rely heavily on scans and photographs, in which object detection can be used to automatically detect signs of sickness in CT or MRI images, such as tumor detection.
- **Industrial Manufacturing**: Object detection can be used in many industrial manufacturing scenarios, like detecting defects in a product line or automating many manual procedures with vision-based robotics.
- **Smart City**: Object detection is also critically important for the smart city initiative. For example, object detection can help monitor the traffic flows from remote sensing images, which helps in avoiding traffic jams and planning better routes.

1.2 Progresses and Challenges

With the recent development of deep neural networks, object detection has experienced significant progress in recent years. Its performance has improved significantly within the last ten years on many challenging generic object detection benchmarks, such as Pascal VOC [5] and MS COCO [6], thanks to the strong feature representation capacity of deep neural networks. Nowadays, most modern object detectors are based on deep neural networks, benefiting from their strong representation capacity.

Despite the encouraging progress achieved, when applied in practical scenarios, object detection still faces a number of challenges, especially under a few constrained scenarios as listed below.

- **Poor Image Quality**: The image quality can be poor in some scenarios like adverse weather or heavy noise interference. In low-quality images, the visual clues of objects, including texture details, contrast, and contours, are largely absent. It is challenging to perform object detection without these visual clues.
- **Limited Computational Resources for Training**: Deep-neural-network-based object detectors usually require long training iterations to produce satisfactory performance. However, the training procedure can be expensive,

time-consuming, environmentally unfriendly, or even unaffordable to many researchers or companies. Under constrained setups with limited resources for training, it is crucial to explore the acceleration of object detectors' training convergence.

- **Limited Number of Training Samples**: Deep neural networks usually require a large amount of annotated data for training. Training over a small amount of data typically leads to overfitting and poor generalization. However, an abundant amount of training samples are often unavailable in many practical scenarios. For example, the samples of rare animals are hard to collect, or the samples for a specific sickness are expensive to label. Since modern object detectors are mostly based on deep neural networks, they will suffer from catastrophic performance drops when trained with only a few samples.

- **Extreme Scale Variation**: Detecting objects of different sizes, especially small objects, has always been a major challenge in object detection. In some cases, the scale variation can be extremely large due to perspectives and distances, adding substantial difficulty to object detection. Moreover, smaller objects also have the lowest detection accuracy on the widely used MS COCO [6] benchmark.

- **Running on Edge Devices with Low Computational Capacity**: Deep neural networks can usually run efficiently on GPUs. However, when deployed to edge devices with limited computational capacity, the complexity of object detectors can be a bottleneck. It is essential to design computationally efficient algorithms for object detectors deployed on computationally limited devices.

- **Domain Gaps**: Deep neural network-based object detectors often suffer significant drops in performance when transferred to different domains, such as variations in camera settings, scenes, weather conditions, or object statistics. Therefore, it is important to improve model's generalization ability to perform robust predictions on various scenarios.

In practical cases, there will exist different constrained scenarios, not limited to the ones listed above. This thesis studies only a few of them under specific constrained cases, which will be detailed in the following subsection.

1.3 Major Contributions

This thesis focuses on accurate and robust object detection under several constrained scenarios, including poor image quality, extreme scale variation, limited computational resources, and limited training samples. Specifically, a number of detection frameworks are developed to tackle the challenges above.

The major contributions of them are stated as follows:

- Chapter 3 studies object detection in optical remote sensing imagery. To address the challenges of detection under the constraint of lack of visual clues (textures and contrast) and large scale variations, the proposed CAD-Net exploits global and local contexts as well as spatial-and-scale-aware attention-modulated features to compensate for information loss and address the new challenges under this scenario.

- Chapter 4 explores the acceleration of the training procedure for the newly proposed DEtection TRansformer (DETR) [1]. Chapter 4 presents a novel semantic-aligned matching strategy to accelerate the training convergence of DETR with improved accuracy. The semantic-aligned matching strategy can further extend to fuse multi-scale features, achieving further boosts in convergence speed and accuracy. Chapter 4 proves that DETR can converge even faster than those conventional ConvNet-based detectors, paving the way for its broader applications.

- Chapter 5 studies few-shot object detection, which aims to detect novel objects with just a few training samples. Chapter 5 presents Meta-DETR, which performs image-level prediction, thus bypassing those low-quality region proposals for novel class objects. Meta-DETR also effectively exploits inter-class correlation among different classes to enhance knowledge generalization to novel classes. Meta-DETR demonstrates competitive detection accuracy even with only a few training samples.

- Chapter 6 explores an pioneering paradigm for Transformer-based object detectors to efficiently use multi-scale features. Chapter 6 presents Iterative Multi-scale Feature Aggregation (IMFA), which exploits multi-scale features only from the most promising and informative locations and significantly improves detection accuracy on multiple Transformer-based object detectors at

marginal costs. The proposed IMFA bridges the gap in applying the newly proposed Transformer-based object detectors to leverage multi-scale features to deliver satisfactory detection accuracy on edge devices with limited computational capacity.

1.4 Outline of the Thesis

This thesis is organized as follows:

- Chapter 1 first introduces the background for the task of object detection, including the motivations and the applications. Then, this chapter points out the latest research progress and the current challenges. Finally, it briefly summarizes our contributions to addressing specific constraints in object detection.
- Chapter 2 systematically reviews the prior arts for generic object detection as well as the commonly used datasets and evaluation metrics for object detection.
- Chapter 3 presents a Context-Aware Detection Network (CAD-Net) for object detection in remote sensing imagery. To adapt to the unique challenges in remote sensing images where objects are densely distributed, arbitrarily oriented, with low contrast and low visual clues, and suffer from heavy noises, the proposed CAD-Net extracts scene-level and object-level contextual information that is highly correlated to objects of interest to compensate the information loss. In addition, a spatial-and-scale-aware attention module is also designed to guide the network to focus on scale-adaptive features and emphasizes the degraded texture details. These special designs enable the proposed CAD-Net to perform robust and accurate object detection even within the challenging remote sensing imagery.
- Chapter 4 studies the acceleration of the training procedure for Transformer-based object detectors. The newly proposed DEtection TRansformer (DETR) has established a fully end-to-end paradigm for object detection with competitive performance. However, DETR suffers from slow training convergence, which hinders its applicability to various detection tasks. Chapter 4 presents

SAM-DETR++ to accelerate the training convergence via a novel semantic-aligned matching mechanism, which effectively reduces the required training resources and simultaneously improves the detection accuracy greatly.

- Chapter 5 presents Meta-DETR for few-shot object detection, where only a few annotated samples are provided for training. Few-shot object detection is of great practical significance as abundant training samples are often unavailable for various reasons, such as sample rarity or expensive labeling labor. Prior arts in few-shot object detection usually adopt R-CNN-based detection frameworks and perform class-wise meta-learning. However, such a paradigm is still constrained by several factors, including *(i)* low-quality region proposals for novel classes and *(ii)* negligence of the inter-class correlation among different classes. These limitations hinder the generalization of base-class knowledge for the detection of novel-class objects. Differently, our proposed Meta-DETR *(i)* is the first image-level few-shot detector, and *(ii)* introduces a novel inter-class correlational meta-learning strategy to capture and leverage the correlation among different classes for robust and accurate few-shot object detection. Meta-DETR works entirely at image level without any region proposals, which circumvents the constraint of inaccurate proposals in prevalent few-shot detection frameworks. In addition, the introduced correlational meta-learning enables Meta-DETR to simultaneously attend to multiple support classes within a single feedforward, which allows to capture the inter-class correlation among different classes, thus significantly reducing the mis-classification over similar classes and enhancing knowledge generalization to novel classes. The proposed Meta-DETR has achieved state-of-the-art performance for few-shot object detection.

- Chapter 6 presents Iterative Multi-scale Feature Aggregation (IMFA) – a novel paradigm to efficiently make use of the multi-scale feature representations in Transformer-based object detectors. Despite their simplicity and good performance, there is yet no generic paradigm for Transformer-based object detectors to efficiently make use of multi-scale features for more accurate detection, largely due to the prohibitive computational cost for the attention mechanism to process high-resolution features. The proposed IMFA sparsely aggregates only the most promising and informative multi-scale features, which improves the performance significantly at a slight computational overhead. IMFA paves the way for the wide application of high-performing

Transformer-based object detectors on edge devices with limited computational capacity.

- Chapter 7 summarizes the contributions of each of our proposed methods in this thesis and then discusses a few future research directions in the field of object detection.

Chapter 2

Generic Object Detection: A Literature Review

This chapter presents the literature review of generic object detection, including the task definition, traditional object detection methods, modern deep-neural-network-based object detection methods, and the datasets and evaluation metrics commonly adopted to evaluate the performance of object detection.

2.1 Overview

Object detection (specifically, 2D image object detection with pre-defined categories) [3, 4] is a fundamental and important task in computer vision. Its goal is to determine whether there are any object instances from a set of pre-defined categories (*e.g.*, humans, faces, dogs, motorbikes, *etc.*) within an image and, if present, to recognize each object instance and predict the location for each object instance using a bounding box. Fig. 2.1 illustrates an example of object detection, where objects of interest (person and horse) are correctly recognized and localized with bounding boxes (image credit: Pascal VOC [5]). Object detection serves as the cornerstone for many other vision tasks, such as scene understanding, tracking, image captioning, person re-identification, and so on. It also has a wide range of applications in autonomous driving, robot vision, surveillance, manufacturing, *etc.*

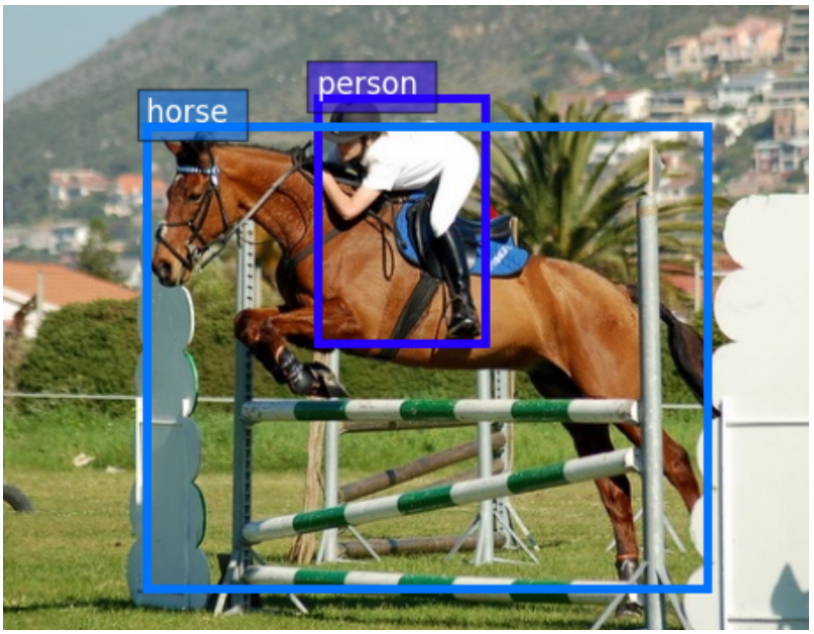


FIGURE 2.1: An example of object detection. Object detection aims to recognize and localize each object of interest in an image using a bounding box.

As a longstanding problem, object detection has been an active research topic for decades, even prior to the rise of deep learning. It is commonly accepted that object detection has gone through two iconic eras [3]: *(i)* the traditional object detection era (before 2014) and *(ii)* the deep-learning-based object detection era (since 2014). Most of the traditional object detectors are developed upon hand-crafted features. In contrast, deep-learning-based object detectors are built upon those deep feature representations that are automatically learned from a large number of training data.

2.2 Traditional Object Detection

In the traditional object detection era, where effective feature representation of images is not available, object detectors have no choice but to rely on sophisticated hand-crafted features [7–9] and heuristics for object detection.

In 2001, the emergence of the Viola-Jones Detector (VJ-Detector) [10] marked the start of the development of traditional object detection. VJ-Detector achieved real-time detection of human faces without any constraints. It is developed upon Haar

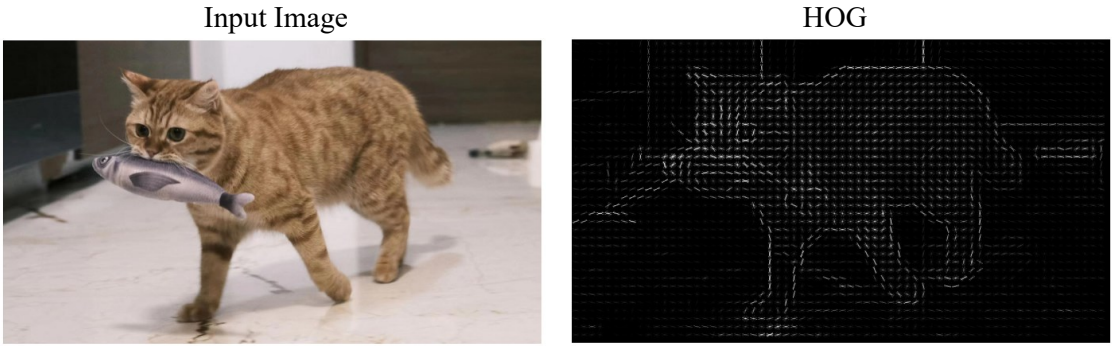


FIGURE 2.2: Visualization of the Histogram of Oriented Gradients (HOG) features.

wavelet features and follows the most simple and straightforward manner for object detection – sliding window. It simply goes through all potential locations and scales to check whether any window contains a human face. To reduce the computational cost to a feasible level, VJ-Detector incorporates three techniques: "Integral Image", "Selective Features", and a multi-stage detection paradigm. Thanks to these techniques, VJ-Detector achieves promising results and runs tens or even hundreds of times faster than its counterparts back then.

In 2005, N. Dalal and B. Triggs proposed Histogram of Oriented Gradients (HOG) feature descriptor [8], which serves as the basis for many HOG-based object detectors [11–13]. Fig. 2.2 illustrates an example of the HOG feature descriptor. HOG counts the occurrences of gradient orientation in localized portions of an image on a dense grid of uniformly spaced cells. Since HOG operates on local cells, it is invariant to geometric and photometric transformations, thus particularly suitable for pedestrian detection within images compared with other feature descriptors.

On top of the HOG feature descriptor [8], Deformable Part-based Model (DPM) was further proposed, marking the peak of the traditional object detection methods. DPM was initially proposed by P. Felzenszwalb [12] in 2008, and further improved by R. Girshick [11, 14, 15]. DPM-based models are the winners for the Pascal VOC-07, -08, and -09 detection challenges [5]. DPM recognizes objects to detect with a mixture graphical model of deformable parts, where training can be deemed as decomposing objects into several parts, and inference can be deemed as ensembling different object parts. In DPM-based models, several important techniques were incorporated, including "hard negative mining", "bounding box regression", and

"context priming", which even have strong influences on modern deep-learning-based object detectors.

However, since 2010, traditional object detection methods have reached a plateau as the performance of hand-crafted features became saturated. Due to the lack of effective and powerful feature representation, most traditional object detectors [10–15] that rely on hand-crafted features fail to deliver satisfactory detection performance under complex scenes and can only work in simple and limited scenarios.

2.3 Deep-Learning-Based Object Detection

In 2012, Krizhevsky *et al.* proposed AlexNet [16], a deep convolutional neural network (CNN or ConvNet), which achieved record-breaking performance in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 image classification challenge. AlexNet [16] demonstrates the advantage of deep neural networks in learning robust and high-level feature representations. Since then, the research focus on computer vision has shifted from traditional methods towards deep-learning-based methods.

Deep-learning-based object detection methods can be categorized into ConvNet-based object detectors and the recently proposed Transformer-based object detectors, which are to be detailed in the following subsections.

2.3.1 ConvNet-Based Object Detection

ConvNet-based object detection emerged in 2014 and has achieved very promising results, becoming the mainstream of object detection. ConvNet-based object detectors can be divided into two types: two-stage methods and single-stage methods.

2.3.1.1 Two-Stage Methods

Two-stage object detectors formulate object detection as a coarse-to-fine prediction process, which first generates a number of object proposals and then refines over each of these proposals.

R-CNN (Regions with CNN features) [17]. R-CNN was proposed in 2014 and is the pioneering work to bring the power of deep neural networks to object detection. It first uses selective search [18] to generate a set of object proposals. Then each object proposal is resized into a fixed-sized image patch to feed into an ImageNet-pretrained CNN model (*e.g.*, AlexNet [16]) for deep feature extraction. Finally, these features of regions are fed to linear SVM classifiers for class prediction as well as bounding box regression.

R-CNN significantly outperforms those DPM-based detectors [11, 12, 14, 15] by large margins. However, R-CNN's drawbacks are obvious. *First*, feature extraction is performed over multiple overlapped regions separately, which is redundant and extremely slow. *Second*, training R-CNN involves multiple separate stages. *Third*, the SVM classifier and bounding box regressor are computationally expensive for training. Later, a series of successive works were proposed to address these drawbacks, which greatly advanced the field of object detection.

SPPNet [19]. R-CNN [17] requires CNN-based feature extraction from thousands of warped image patches separately, which is the bottleneck of the detection pipeline in terms of efficiency. SPPNet introduces spatial pyramid pooling (SPP) into the ConvNet architectures by adding an SPP layer on top of the last convolutional layer to obtain fixed-sized features for each region, which are further fed into the fully connected layers for prediction. SPPNet obtains a significant inference speedup over R-CNN since it only needs to run the CNN feature extraction model on the entire test image once.

Fast R-CNN [20]. In 2015, R. Girshick proposed Fast R-CNN [20] to further improve R-CNN [17] and SPPNet [19]. Fast R-CNN inherits SPPNet's philosophy of sharing feature extraction among regions by introducing a Region of Interest (RoI) pooling layer (RoIPooling) between the last convolutional layer and the fully connected layers to extract fixed-sized features for region proposals. Besides, Fast R-CNN simultaneously learns the CNN model together with the softmax classifier and class-specific bounding box regressor, enabling the end-to-end learning of the object detector. Fast R-CNN further improves the efficiency as well as the detection accuracy by large margins.

Faster R-CNN [21]. The detection speed of Fast R-CNN [20] is still heavily limited by its object proposal generation stage based on selection search [18]. To

address this issue, S. Ren *et al.* proposed Faster R-CNN [21] in 2015. Faster R-CNN's major contribution is a Region Proposal Network (RPN) that generates object proposals with a CNN-based model.

Specifically, RPN first defines k reference boxes (also known as anchor boxes) of different sizes and aspect ratios for each spatial location within the convolutional feature maps. Then, RPN performs classification and bounding box regression over each anchor box, producing a number of object proposals to feed into the Fast R-CNN prediction heads for second-stage classification and bounding box regression. It is noteworthy that RPN and Fast R-CNN heads share the same feature maps produced by the backbone network.

Faster R-CNN [21] is the first end-to-end as well as the first near-realtime deep-learning-based object detector. Since the emergence of Faster R-CNN, it has gone through many design iterations, and its performance has been significantly improved since its original publication. Faster R-CNN is now one of the most widely used object detectors, with numerous variants [22–27] developed on the basis of it.

Feature Pyramid Network (FPN) [22]. Detecting objects of different scales has always been a difficulty in object detection. In 2017, T.-Y. Lin *et al.* proposed Feature Pyramid Network (FPN) based on Faster R-CNN [21]. FPN combines low-resolution but semantically strong features with high-resolution but semantically lacking features using a top-down pathway architecture with lateral connections. FPN significantly advances object detection for objects of various scales with the fusion of multi-scale features, and has become a basic building block for modern state-of-the-art object detectors.

2.3.1.2 Single-Stage Methods

Unlike those two-stage object detectors, single-stage object detectors skip the object proposal stage and directly generate class predictions and bounding box offsets from full images via a single feed-forward process.

YOLO [28–30]. YOLO is the abbreviation of "You Only Look Once", suggesting it only applies a single feed-forward on the full image to produce detection results. YOLO [28] divides the input image into multiple regions and adopts a

ConvNet to predict each region's bounding boxes and class probabilities in one go. Later, R. Joseph also made a series of improvements to YOLO [28] and proposed YOLO v2 [29] and YOLO v3 [30], which benefit from the design of anchor boxes as well as stronger backbone networks for feature extraction.

SSD [31]. SSD is another highly impactful single-stage object detector proposed in 2015. The major contribution of SSD is the introduction of multi-scale features from different layers of the backbone networks to detect objects of different scales. It greatly enhances the detection performance of single-stage object detectors, especially on small objects.

RetinaNet [32]. Despite the simplicity and high efficiency, single-stage detectors still fall behind those two-stage detectors regarding accuracy. T.-Y. Lin *et al.* proposed RetinaNet to bridge this gap in 2017. They discovered that the extreme imbalance of foreground and background is the key root for single-stage detectors' inferior accuracy, and proposed focal loss to replace the standard cross entropy loss for classification. Focal loss effectively highlights those hard and mis-classified examples during training and enables single-stage detectors to have comparable detection accuracy as those two-stage methods.

Single-Stage Anchor-Free Methods [33–35]. Anchor-free object detectors are also worth mentioning, which do not rely on anchor boxes (shape priors) to perform detection. Notably, CornerNet [33] is an object detector that predicts a bounding box as a pair of keypoints (top-left corner and bottom-right corner) using a single convolution neural network. CenterNet [34] and FCOS [35] formulate object detection as a keypoint detection problem, only detecting objects' center points and regressing the bounding boxes from the center locations. These methods are conceptually simple and have also achieved competitive detection performance.

2.3.2 Transformer-Based Object Detection

Transformers [36] were initially proposed by Vaswani *et al.* in 2017 as an attention-mechanism-based framework for natural language processing (NLP) tasks. In 2020, N. Carion *et al.* proposed DEtection TRansformer (DETR) [1] to perform object detection with a Transformer encoder-decoder architecture.

DETR's major difference from ConvNet-based object detectors [21, 35, 37] is that ConvNet-based object detectors usually perform object detection in an indirect way by defining surrogate classification and regression tasks on a large number of anchor boxes or window centers, while DETR directly formulates object detection as a direct set prediction problem. DETR is supervised by a set-based global loss that forces unique predictions via bipartite matching. Therefore, unlike ConvNet-based object detectors that usually require anchor boxes as shape priors and non-maximum suppression (NMS) as post-processing, DETR removes the need for such hand-crafted components and has a simple and straightforward pipeline. DETR [1] has also achieved competitive detection performance and has inspired many follow-up works [38–44].

2.4 Benchmarks and Evaluation Protocols

Modern computer vision techniques are heavily data-driven. Large-scale datasets and appropriate evaluation protocols are the infrastructures for modern computer vision research. This section introduces commonly used benchmarks and evaluation metrics for object detection.

2.4.1 Benchmarks

There have been numerous datasets for object detection. Here, we present the most widely used three datasets for generic object detection in detail.

Pascal VOC [1] **[5].** The Pascal Visual Object Classes (VOC) Challenges is one of the most important competitions in the early years of the computer vision community. The competition was hosted from 2005 to 2012. Pascal VOC object detection dataset contains objects of interest from 20 categories, including person, bird, cat, dog, cow, sheep, car, bus, *etc.* There are two versions of Pascal VOC for object detection: VOC2007 which contains ∼5k training images and ∼12k objects of interest, and VOC2012 which contains ∼11k training images and ∼27k objects of interest. In recent years, the detection performance on Pascal VOC has saturated.

[1] http://host.robots.ox.ac.uk/pascal/VOC/

Therefore, the research community has moved on to more challenging benchmarks for object detection.

ILSVRC[2] **[45].** ILSVRC (ImageNet Large Scale Visual Recognition Challenge) [45] is derived from ImageNet [46], which contains 200 classes of objects. The images in ILSVRC is object-centric. The number of images and objects is significantly higher than Pascal VOC.

MS COCO[3] **[6].** MS COCO is the most commonly used object detection benchmark today. It is also very challenging and contains complex scenes with objects in their natural context. The widely used MS COCO 2017 dataset contains ∼164k images and ∼897k objects of interest from 80 different classes. MS COCO also contains many small objects and densely distributed objects. Now, MS COCO has become the standard for object detection research.

Other than these three most widely used datasets for generic object detection, there are other datasets for generic object detection, including Open Images [47] and LVIS [48], as well as other datasets for detecting specific objects, such as ICDAR [49–52] for scene text detection.

2.4.2 Evaluation Protocols

In recent years, Average Precision (AP) has become the standard metric to evaluate the performance of object detection. AP is usually computed on each category separately and then averaged over all categories.

Some Basic Concepts. Before computing AP, true-positives (TP), false-positives (FP), and false-negatives (FN) should be first counted based on the matching between predicted detections and ground-truth bounding boxes. Note that each ground truth bounding box can only match one TP. Intersection-over-Union (IoU) is used to determine whether a predicted box is matched to a ground truth bounding box. Concretely, TP denotes the number of predicted boxes with IoU larger than a threshold t. FP denotes the number of predicted boxes that do not match any ground truth bounding boxes. FN denotes the missed ground truth

[2] https://image-net.org/challenges/LSVRC/
[3] https://cocodataset.org/

bounding boxes. The IoU threshold t for determining the matching is usually set to 0.5.

With TP, FP, and FN counted, the precision (P) and recall (R) of object detection can be formulated as:

$$P = \frac{TP}{TP + FP}, \tag{2.1}$$

$$R = \frac{TP}{TP + FN}. \tag{2.2}$$

Mean Average Precision (mAP) for Pascal VOC. Pascal VOC [5] adopts mAP@0.5 as the evaluation metric, in which the matching IoU threshold is set to 0.5. mAP is defined as the average detection precision under different recalls averaged by all classes, which can be formulated as:

$$mAP@0.5 = \frac{1}{C}\sum_{i=1}^{C}\int_{0}^{1} P_i(R_i)dR_i, \tag{2.3}$$

where R_i denotes the recall of category i of the detector; $P_i(R_i)$ denotes the precision of category i when the recall of this category is R_i; C denotes total number of classes to be detected.

Average Precision (AP) for MS COCO. MS COCO [6] adopts AP as the evaluation metric, which is defined similarly to Pascal VOC's mAP. One major difference is that instead of using a fixed IoU threshold, MS COCO's AP is further averaged over multiple IoU thresholds (0.5, 0.55, 0.6, ..., 0.9, 0.95). This encourages more accurate localization. Therefore, MS COCO's AP is also referred to as $AP_{0.5:0.95}$. In addition, MS COCO also has other metrics for object detection, including $AP_{0.5}$, $AP_{0.75}$, AP_S, AP_M, and AP_L, where $AP_{0.5}$ measures detection accuracy with coarse localization, $AP_{0.75}$ measures detection accuracy with strict localization. AP_S, AP_M, and AP_L denotes object detection performance for small, medium, and large objects, respectively.

2.5 Object Detection in Constrained Scenarios

Despite the huge progress made, object detection algorithms still face challenges in real-world scenarios. Constrained scenarios, such as low-light conditions, small

object sizes, cluttered backgrounds, different cameras or scenes, limited computational resources, and limited training data, can significantly affect the performance of object detectors.

A significant body of literature has been dedicated to addressing these specific challenges in object detection. For example, [53–55] study domain adaptation, which aims to improve object detectors' performance with domain gaps like different weather conditions and different styles. [56–59] study object detection for rotated objects. [22, 25, 56, 60] study small object detection. [27, 61–66] study few-shot object detection, which aims to detect previously unseen objects with only an extremely limited amount of training samples. [38, 67–69] studies object detection techniques for efficient training and/or inference.

In conclusion, object detection in constrained scenarios is an active ongoing research area that has seen significant advances in recent years. Specific constrained scenarios would require particular techniques. In the following chapters, we study a few typical constrained setups for object detection and develop strategies to improve performance under such constrained scenarios.

Chapter 3

CAD-Net: A Context-Aware Detection Network for Objects in Remote Sensing Imagery [1]

3.1 Introduction

The recent advances in satellites and remote sensing technologies have been leading to a huge amount of high-definition remote sensing images every day that simply goes beyond any manual manipulation and processing. Automated analysis and understanding for remote sensing images have therefore become critically important to make these images useful in many real-world applications such as urban planning, searching, rescuing, environmental monitoring, *etc.* In particular, multi-class object detection, which simultaneously localizes and categories various objects (*e.g.*, planes, vehicles, bridges, roundabouts, *etc.*) within remote sensing images, has become possible due to the increase of sensor resolutions. This challenge goes beyond the traditional scene-level analysis that aims to identify the scene semantics of remote sensing images such as building, grassland, sea, *etc.*, and it has attracted increasing research interest in recent years.

[1] The work in this chapter has been published as Gongjie Zhang, Shijian Lu, and Wei Zhang. "CAD-Net: A Context-Aware Detection Network for Objects in Remote Sensing Imagery." IEEE Transactions on Geoscience and Remote Sensing (**T-GRS**), vol. 57, no. 12, pp. 10015-10024, 2019. (DOI: 10.1109/TGRS.2019.2930982) [26]

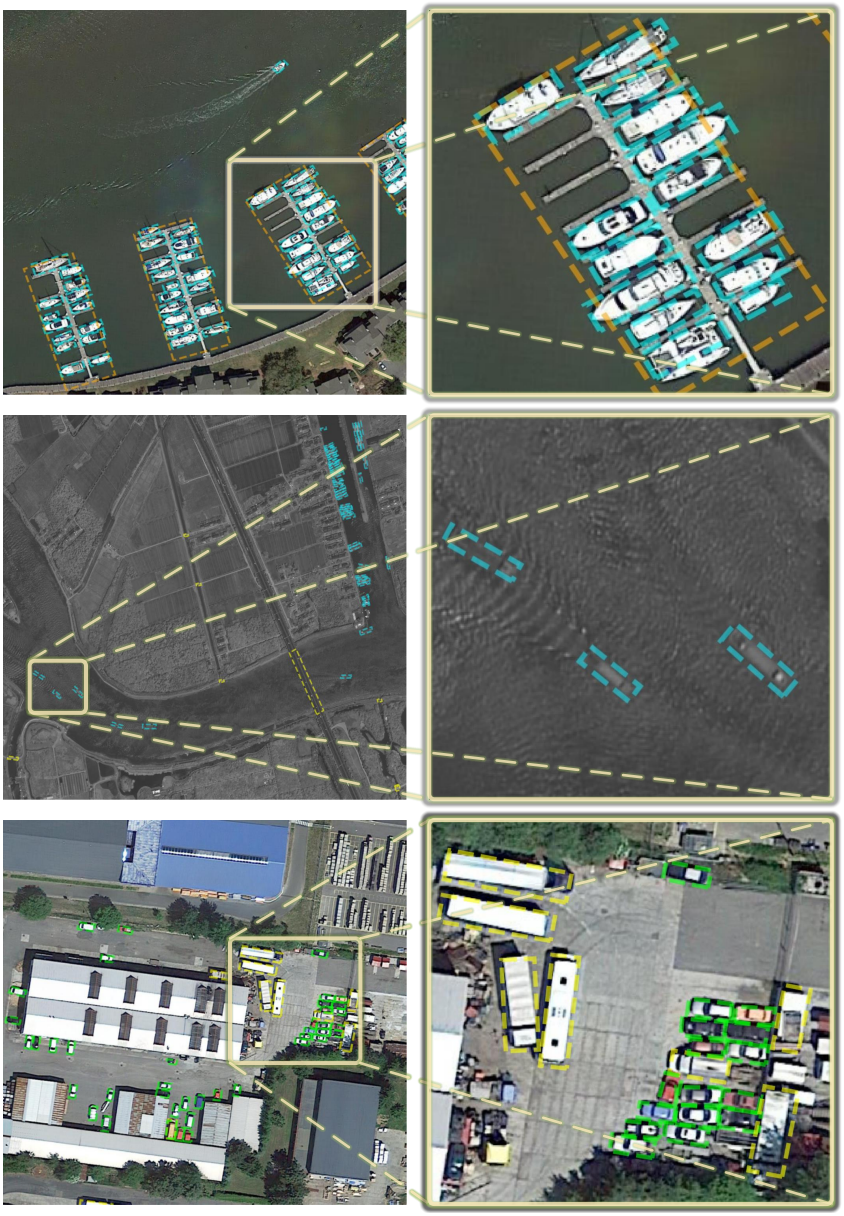


FIGURE 3.1: Illustration of results produced by the proposed CAD-Net for object detection in optical remote sensing imagery. Multi-class objects with different types of image degradation and information loss in colors, contrast and texture are detected and recognized correctly.

The fast development of deep neural networks, especially convolution neural networks (CNNs), has raised the bar of object detection greatly in recent years. A number of CNN-based object detectors [17, 20, 21, 28, 29, 31, 70] have been proposed and very promising results have been achieved over several large-scale object detection datasets such as Pascal VOC [5] and MS COCO [6]. On the other hand, most existing techniques often experience a sharp performance drop while applied to remote sensing images [71], largely due to three factors as illustrated in Fig. 3.1.

First, objects in optical remote sensing images usually lack visual clues such as image contrast and texture details that are critically important to the performance of state-of-the-art object detection techniques. *Second*, objects in remote sensing images are usually densely distributed, appear in arbitrary orientations and have large scale variations, which make object detection an even more challenging task. *Third*, objects captured in optical remote sensing images usually suffer from a large amount of noises due to various interference while light gets reflected and travels a long way back to satellite sensors.

In this chapter, we specially design a *Context-Aware Detection Network (CAD-Net)* for object detection in optical remote sensing images. Fig. 3.2 illustrates the overall framework of the proposed CAD-Net, which is developed on top of the commonly used Faster R-CNN [21]. As Fig. 3.2 shows, CAD-Net consists of a *Global Context Network (GCNet)* that learns the correlation between interested objects and their corresponding global scenes, *i.e.*, the correlation between features of objects and features of the whole image. The GCNet is inspired by the observations that optical remote sensing images usually cover large areas where the scene-level semantics often provides important clues on both object locations and object categories, *e.g.*, ships often appear in seas/rivers, helicopters hardly appear around residence areas, *etc.* In addition, CAD-Net consists of a *Pyramid Local Context Network (PLCNet)* that learns multi-scale co-occurrence features and/or co-occurrence objects surrounding the objects of interest. Compared with images captured by ground-level sensors, remote sensing images from the top view often contain richer and more distinguishable co-occurrence features and/or objects that are very useful for object category and object position reasoning, *e.g.*, vehicles appearing around each other, ships in harbors, bridges above rivers, *etc.* Furthermore, a *Spatial-and-Scale-Aware Attention Module* is designed to guide the network focus on more informative contextual regions at the appropriate image scales.

The main contributions of this chapter are fourfold. *First*, it designs an innovative context-aware detection network to adapt to the constrained scenarios of object detection in optical remote sensing images by effectively learning global and local contexts. To the best of our knowledge, this is the first work to incorporate global and local contextual information for object detection in optical remote sensing images. *Second*, it designs a spatial-and-scale-aware attention module that guides the network to focus on more informative regions at the appropriate image feature

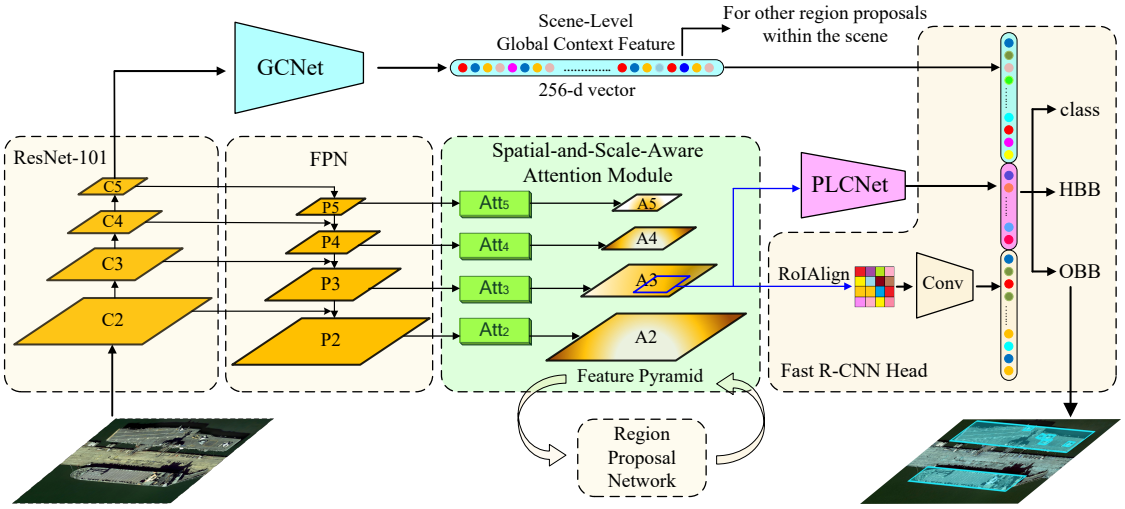


FIGURE 3.2: The overall framework of the proposed CAD-Net. On top of the basic architecture of Faster R-CNN with Feature Pyramid Network (FPN) (in beige), a Global Context Network (GCNet, highlighted in cyan) and a Pyramid Local Context Network (PLCNet, highlighted in purple) are designed to capture global contexts at the scene level and local contexts at the object level, respectively. Besides, a Spatial-and-Scale-Aware Attention Module (highlighted in light green) is designed to guide the network to focus on more informative regions at the appropriate feature scales while suppressing irrelevant information. In addition to standard horizontal bounding box (HBB) regression to predict axis-aligned bounding boxes, an oriented bounding box (OBB) regression branch is added to produce OBB results, which better align with arbitrarily oriented objects in remote sensing images.

scales. *Third*, it verifies the uniqueness of remote sensing object detection, and also provides an insightful and novel solution to bridge the gap with respect to object detection from images captured by ground-level sensors. *Fourth*, without bells and whistles, it develops an end-to-end trainable detection network – CAD-Net, which obtains state-of-the-art performance over two challenging object detection datasets for optical remote sensing images.

3.2 Methodology

3.2.1 Overview

The overall framework of our proposed Context-Aware Detection Network (CAD-Net) is illustrated in Fig. 3.2. As the figure shows, CAD-Net is developed upon the structure of a classical two-stage object detector – Faster R-CNN [21] with Feature

Pyramid Network (FPN) [22]. To adapt to the constrained setups for object detection in optimal remote sensing images, a Global Context Network (GCNet) and a Pyramid Local Context Network (PLCNet) are designed and fused to extract contextual information at global scene level and local object level, respectively. In addition, a spatial-and-scale-aware attention module is designed, which guides the network to focus on more informative regions as well as more appropriate image feature scales. All designed components are off-the-shelf and can be incorporated into existing modern object detection networks without any adaptation and extra supervision information. More details are to be discussed in the ensuing subsections.

Given an image $\mathcal{I}$ and a region proposal $\mathcal{P}$, the detection of the objects $\mathcal{O}_{\mathcal{P}}$ with respect to $\mathcal{P}$ can be formulated as:

$$\mathcal{O}_{\mathcal{P}} = Det(\mathcal{I}, \mathcal{P}), \tag{3.1}$$

where $Det(\cdot)$ denotes the procedure of joint object classification and bounding box regression. In the widely adopted region-based object detection approaches, Eq. 3.1 is often approximated by RoIPooling [20] that guides the network to focus on the proposal region and ignore the rest parts of the image. The new formulation can thus be presented by:

$$\mathcal{O}_{\mathcal{P}} = Det[\Psi(\mathcal{P}, \mathcal{I}), \mathcal{P}], \tag{3.2}$$

where $\Psi(\cdot)$ denotes the RoIPooling operation.

The approximation in Eq. 3.2 is based on the assumption that all useful information for a specific region $\mathcal{P}$ lies within the region itself. This assumption works for most images from ground-level sensors where discriminative object features are usually captured and kept well. However, discriminative object features such as edges and texture details are often severely degraded for optical remote sensing images due to various noises and information loss. Under such circumstances, global and local contexts that are often correlated with objects of interest closely become important and should be incorporated to compensate the feature degradation and information loss. The incorporation of the global and local contexts can thus be formulated as follows:

$$\mathcal{O}_{\mathcal{P}} = Det\left\{[\Psi(\mathcal{P}, \mathcal{I}); G(\mathcal{I}); L(\mathcal{P}, \mathcal{I})], \mathcal{P}\right\}, \tag{3.3}$$

where $G(\cdot)$ denotes GCNet for obtaining global contextual features, $L(\cdot)$ denotes PLCNet for obtaining local contextual features, and $(\cdot;\cdot)$ denotes the operation of concatenation.

3.2.2 Global Context Network

Remote sensing images usually capture a large area of land that carries strong semantic information that characterizes the captured scene. In addition, the semantics of the captured scenes are often closely correlated with objects within the scenes, *e.g.*, sea *vs.* ship, airport *vs.* airplane, *etc.* With these observations, we design a Global Context Network (GCNet) that learns the global scene semantics and uses them as certain priors for better detection of objects in remote sensing images. More specifically, GCNet learns the correlation between a scene and objects within the scene and use the learned correlation as certain global contexts to compensate the loss of discriminative object features. The proposed GCNet can be formulated as follows:

$$G(\mathcal{I}) = \psi\left\{\Phi_G\left[\Lambda(\mathcal{I})\right]\right\}, \tag{3.4}$$

where $\Lambda(\mathcal{I})$ denotes the final feature maps of the feature extraction network, i.e. the C5 level of the ResNet-101 backbone as illustrated in Fig. 3.2, $\Phi_G(\cdot)$ is implemented by a stack of convolutional neural network layers that extracts global features, and $\psi(\cdot)$ denotes a pooling operation that squeezes the spatial channels of the feature maps into a vector which helps to suppress the sensitivity to scale variations. We empirically adopt global average pooling for $\psi(\cdot)$ in our implemented system.

3.2.3 Spatial-and-Scale-Aware Attention Module

Before introducing Pyramid Local Context Network (PLCNet), we first introduce the proposed Spatial-and-Scale-Aware Attention Module, since features for both region proposals and object-level contexts are extracted from feature pyramids generated by this attention mechanism, as illustrated in Fig. 3.2.

Visual attention has been proven very useful in different computer vision tasks such as image captioning [72], person re-identification [73], *etc.* The idea is inspired by the human visual system that does not process an entire image at one go but tends

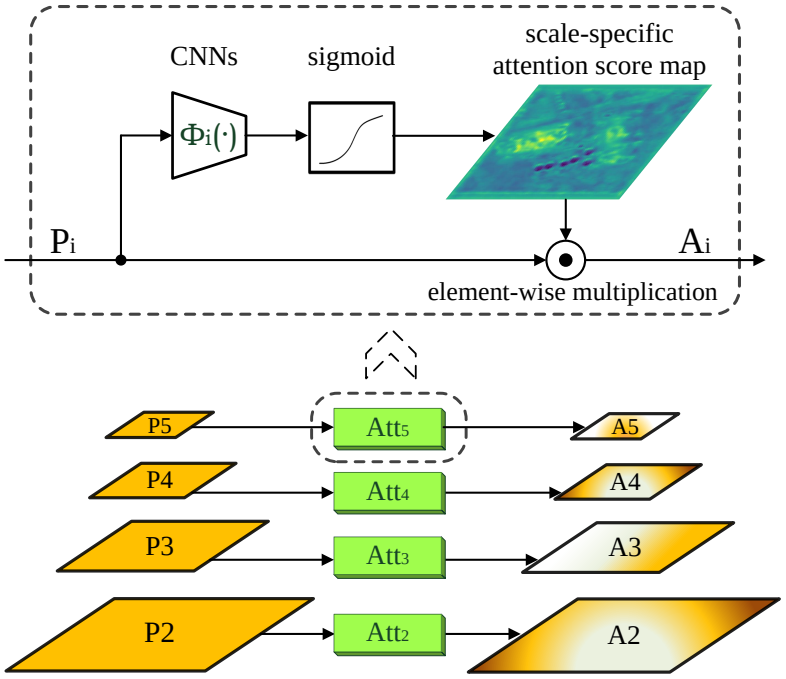


FIGURE 3.3: The architecture of our proposed spatial-and-scale-aware attention module. Scale-specific spatial attention is applied to different feature scales, guiding the network to focus on informative regions at appropriate feature scales while suppressing irrelevant information.

to focus on more informative regions sequentially. Here, to adapt to the constrained setups in remote sensing images where there exists huge object scale variation and a lack of visual clue, we design a spatial-and-scale-aware attention module that learns to adaptively focus on more prominent regions (spatial-aware attention) at relevant scales of feature maps (scale-aware attention). The spatial-aware attribute helps the network to handle objects with sparse texture and low contrast with the background, and the scale-aware attribute helps to handle objects of very different scales. The combination of both significantly facilitates the learning of object detection for optical remote sensing images.

The proposed spatial-and-scale-aware attention module takes as input the FPN-generated feature pyramid that contains feature maps $P_2 - P_5$, as illustrated in Fig. 3.3. For a feature map corresponding to one specific scale $P_i (i \in [2, 3, 4, 5])$, an attention-modulated feature map A_i is determined as follows:

$$S_i = \sigma[\Phi_i(P_i)], \tag{3.5}$$

$$A_i = S_i \odot P_i, \tag{3.6}$$

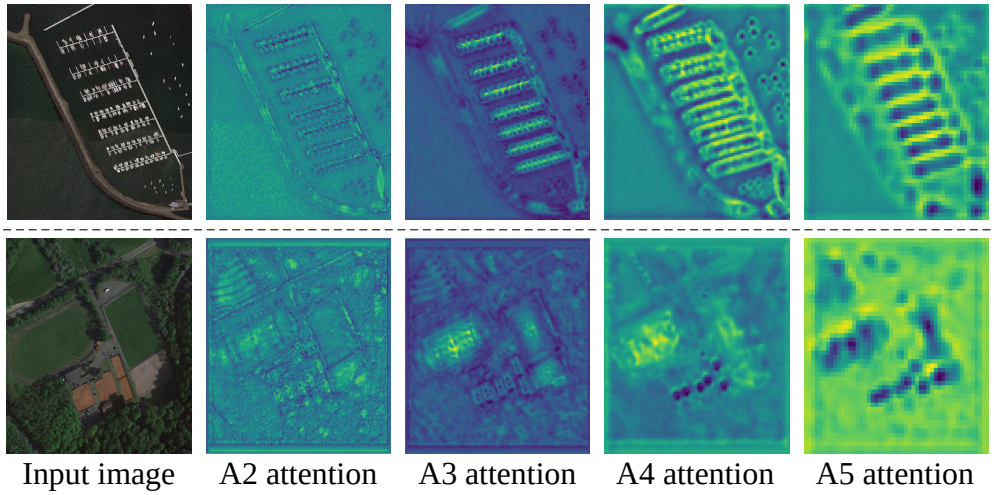


FIGURE 3.4: Illustration of our proposed spatial-and-scale-aware attention responses at different feature scales. Brighter regions indicate higher attention responses. The proposed spatial-and-scale-aware attention module is capable of focusing on informative regions at the appropriate feature scales while suppressing responses in irrelevant and noisy areas.

where $\sigma(\cdot)$ denotes the sigmoid function, S_i is the attention map of the i-th feature map, A_i is the i-th attention-modulated feature map, and $\odot$ denotes element-wise multiplication. The attention map computation $\Phi_i(\cdot)$ is accomplished by a stack of convolutional neural network layers. Note that a separate $\Phi_i(\cdot)$ is implemented to compute each scale-specific attention map. This design ensures that our proposed attention module is both spatial-aware and scale-aware, enabling it to focus on more informative regions at appropriate scales with irrelevant information effectively suppressed.

Fig. 3.4 presents the attention response maps that are generated by the proposed spatial-and-scale-aware attention module. As Fig. 3.4 shows, our proposed attention module is not only spatial-aware, but also scale-aware, which can selectively focus on more informative regions for features from different scales. For example, small-scale ships get stronger responses at lower feature scales A2 and A3 (as shown in Fig. 3.3) that capture more detailed information, while large-scale harbors get stronger responses at higher feature scales A4 and A5 that capture more high-level information as illustrated in the first sample image. In addition, our attention module is able to guide the network to focus on useful texture details that are degraded by noises, *e.g.*, the skeleton of the harbors in the first sample image and the middle line of the court in the second sample image.

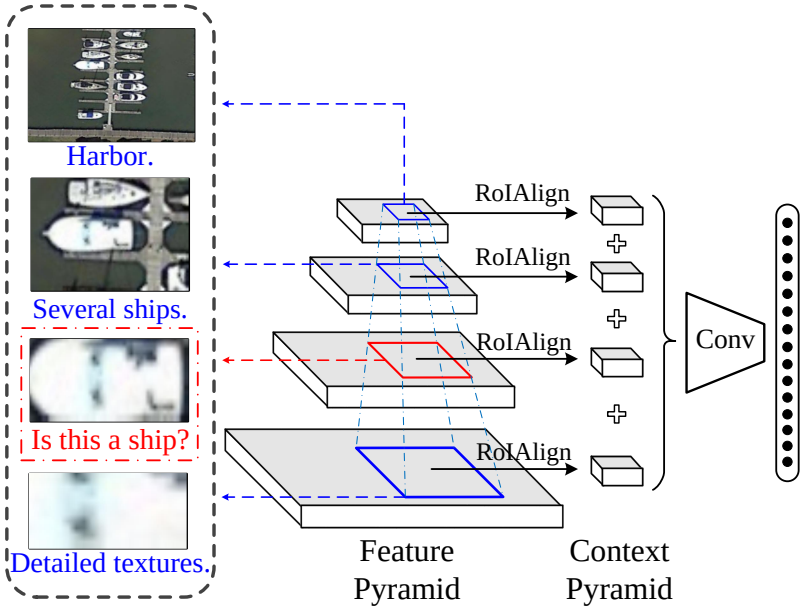


FIGURE 3.5: Illustration of the framework of our proposed Pyramid Local Context Network (PLCNet). PLCNet is able to extract features of the proposal region from different feature scales and capture their correlations to serve as supplementary information for object detection.

3.2.4 Pyramid Local Context Network

Besides global contexts, local contexts that characterize the correlation between an object and its neighboring objects and/or features also capture useful information and can be exploited to compensate the information loss in optical remote sensing images. Based on the observation that both objects and their local contexts in attention-modulated feature maps $A_i (i \in [2, 3, 4, 5])$ are scale-sensitive, we design a Pyramid Local Context Network (PLCNet) to capture the object/feature correlation between objects and their local contexts, as illustrated in Fig. 3.5.

Given a region proposal $\mathcal{P}$ (*e.g.*, the ship proposal in the red-colored box in Fig. 3.5), a set of local contexts of the corresponding region from different scales are employed to learn the cross-scale local contexts around $\mathcal{P}$ as illustrated in Fig. 3.5. A context pyramid is designed, which first extracts and concatenates pooled features from different scales and then fuses the concatenated features through convolutional neural networks (*i.e.*, the *Conv* in Fig. 3.5). The fused features are finally concatenated with the region features as well as the aforementioned global context features for classification and bounding box regression.

As illustrated in Fig. 3.5, even humans will find it challenging to tell whether the proposed region (highlighted in red-colored box) is a ship by just focusing on the proposed region itself. Under such circumstance, the local contexts from different scales (*e.g.*, cluster of ships and harbors as shown in Fig. 3.5) will provide strong clues that the region proposal is very likely to be a ship. Our proposed PLCNet is trained to learn from such correlated features and/or objects, which often help a lot in the presence of sparse texture, low contrast, and severe information loss in optical remote sensing images.

3.2.5 Network Optimization

3.2.5.1 Target Generation for Oriented Objects

Unlike generic object detection where objects of interest can be well localized by horizontal bounding boxes (HBB), objects in optical remote sensing images are typically densely distributed and arbitrarily oriented. To adapt to the unique scenarios for object detection in remote sensing images, the proposed CAD-Net adopts both horizontal bounding boxes (HBB) and oriented bounding boxes (OBB) as training targets, so the predicted bounding boxes can align better with the objects in remote sensing images, as illustrated in Fig. 3.1.

Concretely, similar to generic object detection [17, 20, 21, 28, 29, 31], each object can be formulated into HBB format as:

$$HBB : \{x_{min}, y_{min}, x_{max}, y_{max}\}, \tag{3.7}$$

where $\{x_{min}, y_{min}\}$ denotes the coordinates for the bounding box's top-left corner, and $\{x_{max}, y_{max}\}$ denotes the coordinates for the bounding box's bottom-right corner. Besides, each object can also be formulated into OBB format as:

$$OBB : \{x_{center}, y_{center}, w, h, \theta\}, \tag{3.8}$$

where $\{x_{center}, y_{center}\}$ denotes the center coordinates of the bounding box, $\{w, h\}$ denotes the width and height of the bounding box, and θ denotes the orientation that lies within $[0, 90^{\circ})$ to ensure that each object only has a single legitimate ground truth. During the training procedure of CAD-Net, the OBB training targets

as defined in Eq.(3.8) are generated by a set of rotated rectangles that best overlap with the provided quadrilateral annotations.

3.2.5.2 Loss Function

We adopt loss function following standard Faster R-CNN [21] with FPN [22]. No extra loss function is required for training our introduced components. Loss functions for the proposed CAD-Net can be formulated as:

$$\mathcal{L} = \mathcal{L}_{rpn} + \mathcal{L}_{head}, \tag{3.9}$$

where $\mathcal{L}_{rpn}$ denotes loss for the Region Proposal Networks (RPN), and $\mathcal{L}_{head}$ denotes loss for the Fast R-CNN head. They are formulated as:

$$\begin{aligned} \mathcal{L}_{rpn} = \lambda_1 \frac{1}{N_{cls}} \sum_i \mathcal{L}_{cls}(p_i, p_i^*) \\ + \lambda_2 \frac{1}{N_{reg}} \sum_i p_i^* \mathcal{L}_{reg}(t_i, t_i^*), \end{aligned} \tag{3.10}$$

$$\begin{aligned} \mathcal{L}_{head} = \lambda_3 \frac{1}{K_{cls}} \sum_i \mathcal{L}_{cls}(c_i, c_i^*) \\ + \lambda_4 \frac{1}{K_{reg}} \sum_i [c_i^* \geq 1] \mathcal{L}_{reg}(HBB_i, HBB_i^*) \\ + \lambda_5 \frac{1}{K_{reg}} \sum_i [c_i^* \geq 1] \mathcal{L}_{reg}(OBB_i, OBB_i^*). \end{aligned} \tag{3.11}$$

Here, L_{cls} is implemented by cross-entropy loss. L_{reg} is implemented by smooth-L1 loss. p_i is the RPN predicted probability of anchor i being an object of interest. p_i^* is the target objectness score assigned to anchor i following the assignment rule of Faster R-CNN [21] with FPN [22]. t_i^* is the target bounding box regression offset for positive anchor i. t_i is the predicted regression offset for anchor i. N_{cls} and N_{reg} denote the total and positive number of anchors, respectively. c_i^* is the ground truth class. c_i is the discrete probability distribution for the predicted classes. HBB_i, OBB_i are predicted regression offsets for horizontal/oriented bounding box. HBB_i^*, OBB_i^* are target regression offsets. K_{cls} and K_{reg} denote the total and positive number of region proposals, respectively. $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$ are all set to 1.0 in our experiments.

3.3 Experiments

This section presents experimentation, including datasets and evaluation metrics, implementation details, experimental results over two public remote sensing object detection datasets, and an ablation study of the proposed CAD-Net.

3.3.1 Datasets

The proposed method is evaluated over two publicly available datasets for object detection in remote sensing imagery.

- **DOTA [71]:** DOTA is a recently published large-scale open-access dataset for benchmarking object detection in remote sensing imagery. It contains 2,806 aerial images captured using different sensors and platforms, in which over 188,000 object instances are annotated using quadrilaterals. The images from DOTA are diverse in sizes, ground sample distances (GSDs), sensor types, *etc.*, and the captured objects also exhibit rich variation in term of scales, shapes and orientations. 15 categories of objects are annotated, which include plane (PL), baseball diamond (BD), bridge, ground track field (GTF), small vehicle (SV), large vehicle (LV), ship, tennis court (TC), basketball court (BC), storage tank (ST), soccer ball field (SBF), roundabout (RA), harbor, swimming pool (SP) and helicopter (HC). This dataset is divided into three subsets for training (1/2), validation (1/6), and test (1/3), respectively, where the ground truth of the test set is not publicly accessible. Detection performance on the test set can be obtained by submitting detection results to the dataset organizer's server.

- **NWPU-VHR10 [74, 75]:** NWPU-VHR10 is a publicly accessible dataset for object detection in remote sensing images. It contains 800 very-high-resolution remote sensing images in total, among which 650 are positive and 150 are negative without containing any interested objects. The dataset has annotations of 10 types of objects, including plane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge and vehicle. All objects of interest are annotated using horizontal bounding boxes (HBB) that are publicly accessible.

3.3.2 Evaluation Metric

We adopt the mean Average Precision (mAP) with an Intersection-over-Union (IoU) threshold of 0.5 (mAP@0.5) as the evaluation metric in our experiments, as mAP@0.5 is well-defined and has been widely adopted for evaluation of multi-category object detection performance in the literature. mAP can be generally formulated as:

$$mAP = \frac{1}{C}\sum_{i=1}^{C}\int_{0}^{1} P_i(R_i)dR_i, \tag{3.12}$$

where R_i denotes the recall of category i of the detector; $P_i(R_i)$ denotes the precision of category i when the recall of this category is R_i; C denotes total number of classes to be detected. It is able to well reflect the global performance of multi-category object detection. In our experiments, the exact definition of mAP@0.5 aligns with the metric for Pascal VOC 2012 object detection challenge [5].

3.3.3 Implementation Details

Data Pre-Processing: Optical remote sensing images often have huge image sizes, *e.g.*, the size of DOTA images can be up to 6,000 × 6,000 pixels. To fit the hardware memory in the training stage, we crop images into patches of size 1,600 × 1,600 pixels with an overlap of 800 pixels among neighboring patches. In the inference stage, image patches of 4,096 × 4,096 pixels are cropped from test images with an overlap of 1,024 pixels among neighboring patches. Zero padding is applied if an image is smaller than the cropped image patches. Note that cropping images into patches does not reduce the image data. It just divides a large remote sensing image into multiple small patches so that each cropped image patches can be stored and processed by GPU. Other standard pre-processing processes are also performed, such as global contrast normalization.

Network Setups: We adopt the ResNet-101 [76] as network backbone for feature extraction. As a common practice, this ResNet-101 is pre-trained on the ImageNet [46] and then fine-tuned over the training images of the two studied remote sensing image datasets during our training procedure. As objects in remote sensing images are often arbitrarily oriented, our proposed CAD-Net is designed to produce both HBB and OBB detection predictions simultaneously, provided that the OBB

TABLE 3.1: Object detection results of CAD-Net and comparison with state-of-the-art object detectors on the test set of DOTA dataset. Methods specifically designed for remote sensing images are marked with △ for fair comparison.

Method	data	mAP	PL	BD	bridge	GTF	SV	LV	ship	TC	BC	ST	SBF	RA	harbor	SP	HC
YOLO_v2 [29, 71]	T+V	25.5	52.8	24.2	10.6	35.5	14.4	2.4	7.4	51.8	44.0	31.4	22.3	36.7	14.6	22.6	11.9
SSD [31, 71]	T+V	17.8	41.1	24.3	4.6	17.1	15.9	7.7	13.2	40.0	12.1	46.9	9.1	30.8	1.4	3.5	0.0
R-FCN [70, 71]	T+V	30.8	39.6	46.1	3.0	38.5	9.1	3.7	7.5	42.0	50.4	67.0	40.3	51.3	11.1	35.6	17.5
FR-H [21, 71]	T+V	40.0	49.7	64.2	9.4	56.7	19.2	14.2	9.5	61.6	65.5	57.5	51.4	49.4	20.8	45.8	24.4
FR-O [21, 71]	T+V	54.1	79.4	77.1	17.7	64.1	35.3	38.0	37.2	89.4	69.6	59.3	50.3	52.9	47.9	47.4	46.3
△ R-DFPN [56, 60]	T+V	57.9	80.9	65.8	33.8	58.9	55.8	50.9	54.8	90.3	66.3	68.7	48.7	51.8	55.1	51.3	35.9
△ Yang et al. [60]	T+V	62.3	81.3	71.4	36.5	67.4	61.2	50.9	56.6	90.7	68.1	72.4	55.1	55.6	62.4	53.4	51.5
△ Azimi et al. [77]	T	65.0	81.2	68.7	43.4	61.1	65.3	67.7	69.2	90.7	71.5	70.2	**55.4**	57.3	66.5	61.3	45.3
△ Azimi et al. [77]	T+V	68.2	81.4	74.3	47.7	70.3	64.9	**67.8**	70.0	90.8	79.1	**78.2**	53.6	**62.9**	**67.0**	64.2	50.2
△ CAD-Net(Ours)	T	67.4	**88.3**	71.7	**51.4**	66.5	**72.4**	64.5	**76.7**	90.8	77.3	74.2	45.9	60.2	65.7	56.7	48.3
△ CAD-Net(Ours)	T+V	**69.9**	87.8	**82.4**	49.4	**73.5**	71.1	63.5	**76.7**	**90.9**	**79.2**	73.3	48.4	60.9	62.0	**67.0**	**62.2**

ground truth is available. Concretely, for DOTA dataset [71], CAD-Net generates both HBB results and OBB results. For NWPU-VHR10 dataset [74, 75], CAD-Net only generates HBB results, as OBB ground truth is not provided by this dataset.

Training Setups: We adopt the stochastic gradient descent (SGD) with momentum for network optimization. Our model is trained on a single NVIDIA Tesla P100 SXM2 GPU with 16 GB memory, along with the deep learning framework PyTorch. Batch size is set to 1. Total training iterations for DOTA and NWPU-VHR10 are 130,000 and 30,000, which take around 36 and 6 hours, respectively.

3.3.4 Experiment Results

Table 3.1 presents CAD-Net's object detection results and its comparison with state-of-the-art methods on the test set of DOTA dataset [71]. Note that all methods listed in Table 3.1 use ResNet-101 [76] as the backbone network, except that YOLO v2 [29, 71] and SSD [31, 71] adopt GoogLeNet [78] and Inception v2 [79], respectively. As Table 3.1 shows, our proposed CAD-Net outperforms the Faster R-CNN baseline (FR-O in Table 3.1) [21, 71] by a large margin (+ 15.8% mAP), demonstrating its effectiveness in adapting to the constrained scenarios of object detection in remote sensing images. In addition, it outperforms state-of-the-art performance by ∼2% under two data split setups ('T' means only training images are used in training, and 'T+V' means both training images and validation images are used in training). Besides, we highlight that Azimi's method [77] adopts Inception module [78], deformable convolution [80], online hard example mining

TABLE 3.2: Object detection results of CAD-Net and comparison with state-of-the-art object detectors on the NWPU-VHR10 dataset.

Method	**mAP**	**PL**	**ship**	**ST**	**BD**	**TC**	**BC**	**GTF**	**harbor**	**bridge**	**vehicle**
COPD [74]	54.9	62.3	69.4	64.5	82.1	34.1	35.3	84.2	56.3	16.4	44.3
Transferred CNN [16]	59.6	66.0	57.1	85.0	80.9	35.1	45.5	79.4	62.6	43.2	41.3
RICNN [82]	73.1	88.7	78.3	86.3	89.1	42.3	56.9	87.7	67.5	62.3	72.0
Faster R-CNN [21]	84.5	90.9	86.3	90.5	**98.2**	89.7	69.6	**100**	80.1	61.5	78.1
Li et al. [83]	87.1	**99.7**	**90.8**	90.6	92.9	**90.3**	80.1	90.8	80.3	68.5	87.1
CAD-Net (Sep. 1)	92.1	90.9	80.8	96.4	90.9	90.2	90.9	99.6	100	90.9	89.9
CAD-Net (Sep. 2)	91.3	100	63.4	99.7	99.1	81.8	90.9	99.7	100	88.7	89.8
CAD-Net (Sep. 3)	91.0	100	89.6	90.6	90.8	90.8	79.6	99.5	100	79.0	90.0
CAD-Net (Avg.)	**91.5**	97.0	77.9	**95.6**	93.6	87.6	**87.1**	99.6	**100**	**86.2**	**89.9**

(OHEM) [81], multi-scale training and inference, *etc.*, whereas we target a clean and efficient model with outstanding performance. Our model should be able to achieve higher detection accuracy by including those well-proven performance-boosting components.

We also evaluate the proposed CAD-Net on the NWPU-VHR10 dataset [74, 75] and benchmark it with state-of-the-art methods. Since the NWPU-VHR10 dataset [74, 75] does not specify the partition for training and testing sets, we randomly choose 75% of the positive images as training set and the rest positive images as test set following the widely adopted partition scheme [83], except that we do not include any negative images for training. Table 3.2 presents CAD-Net's object detection results and its comparison with state-of-the-art methods. Here, we provide experimental results of CAD-Net on 3 random separations of NWPU-VHR10 dataset to provide more compelling results. As Table 3.2 shows, the proposed CAD-Net also obtains superior object detection performance as compared with the state-of-the-art methods.

Fig. 3.6 shows a few sample images from the DOTA dataset [71] and the corresponding detection results by using the baseline model – Faster R-CNN [21] with FPN [22] (the left column) and our proposed CAD-Net (the right column). As Fig. 3.6 shows, Faster R-CNN with FPN still tends to produce incorrect detection results under different adverse scenarios, such as ships in the first sample image (mis-detected as large vehicles), storage tanks of different styles in the second sample image (mis-detected as roundabouts), harbors occluded by ships and ships with little texture detail in the third sample image (mis-detected as false negatives), and

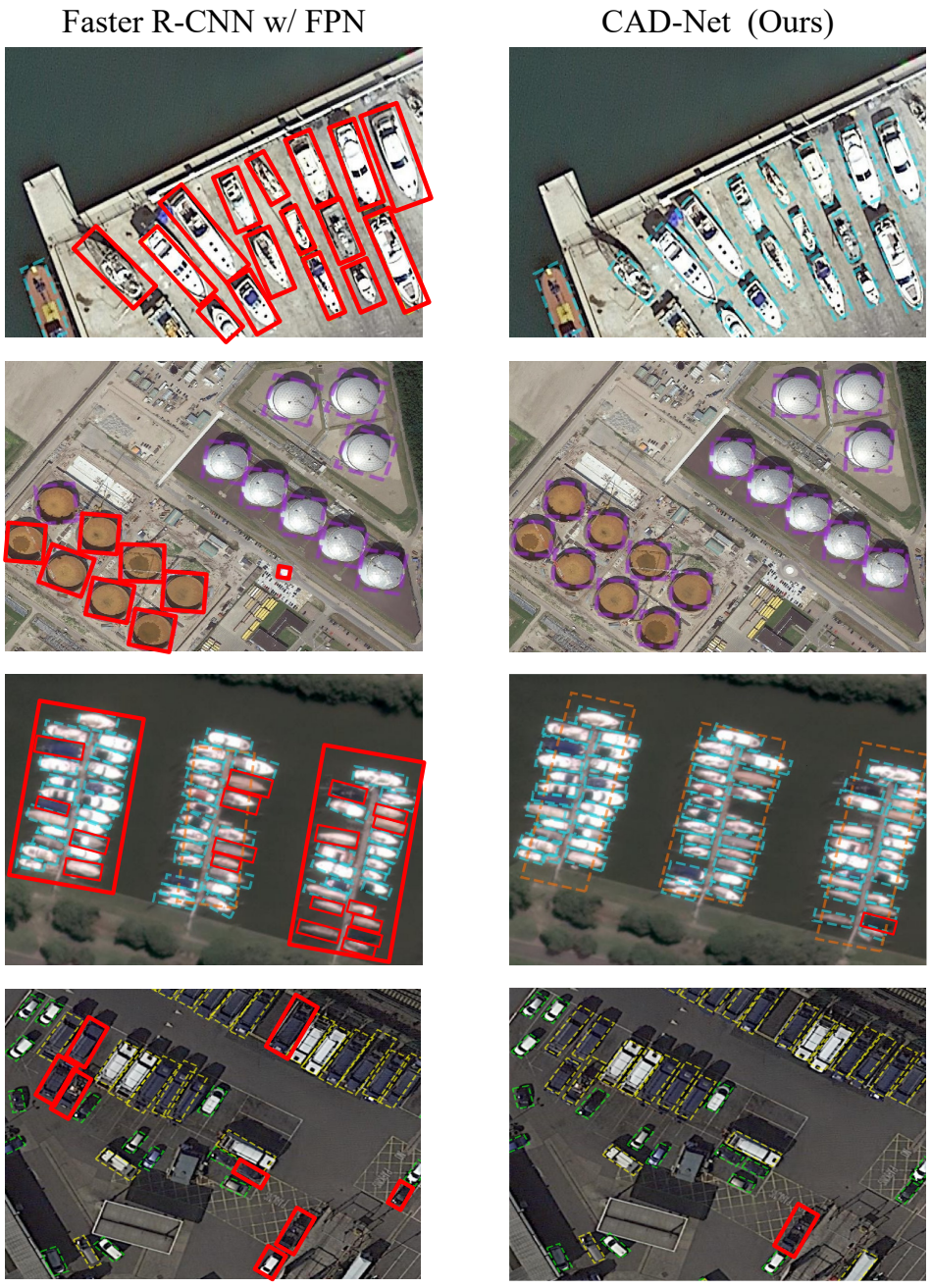


FIGURE 3.6: Illustration of object detection results within optical remote sensing images. The left column shows detection results by the baseline Faster R-CNN with FPN. The right column shows detection results by our proposed CAD-Net, in which most objects are detected correctly. Dashed boxes of different colors show true positive detection results of objects of different categories. Red-colored boxes denote false positives or false negatives. The detection results show that our proposed CAD-Net is tolerant to different types of degradation and information loss. All sample images are from the DOTA dataset.

vehicles with very low contrast with the background in the fourth sample image

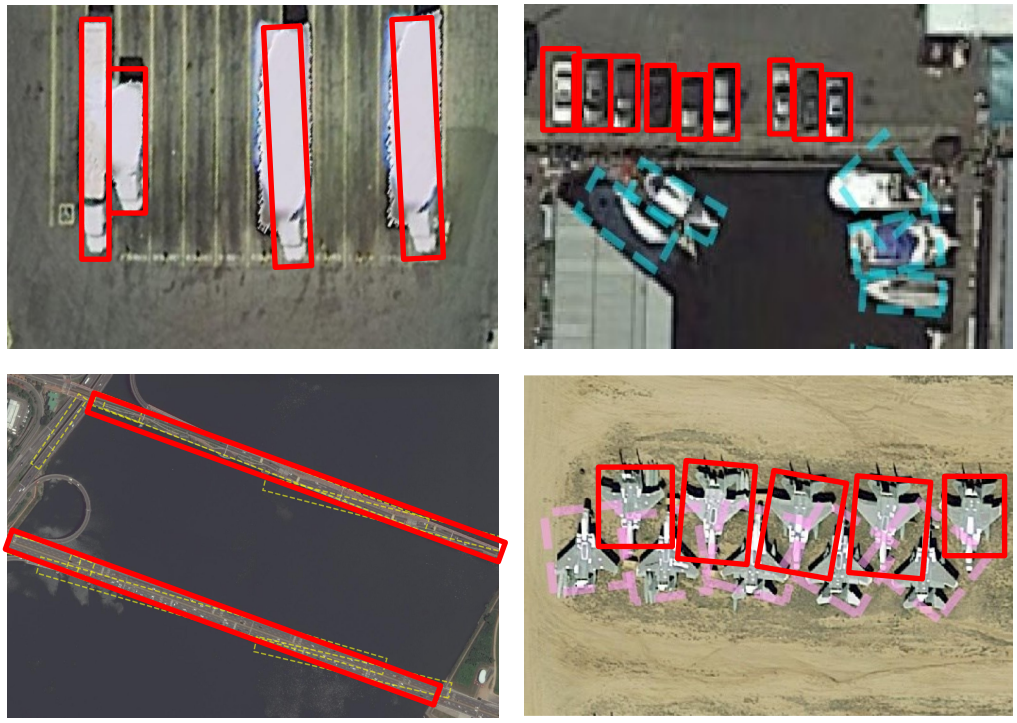


FIGURE 3.7: Failure cases of our proposed CAD-Net. Dashed boxes of different colors show true positive detection results of objects of different categories. Red-colored boxes denote false positives or false negatives. All sample images are from the DOTA dataset.

(mis-detected as false negatives). As a comparison, the proposed CAD-Net is capable of correctly detecting those objects under various adverse scenarios as illustrated in the right column of Fig. 3.6. The superior detection performance is largely attributed to the inclusion of the global contexts, local contexts, spatial-and-scale-aware attention, strong and balanced semantics information, and accurate rotation angle regression within the proposed CAD-Net.

On the other hand, the proposed CAD-Net are still prone to some detection failures under several typical situations, as illustrated in Fig. 3.7. First, the proposed CAD-Net is sensitive to strong light interference, largely due to the lack of relevant training images within the training set. Second, CAD-Net often produces missed detection for small vehicles even when the small vehicles have good visual quality. We highly suspect that this is due to the inaccurate annotation of the training images. In particular, many small vehicles are not annotated, probably because of a huge amount of small vehicles in images and limited manpower. Third, CAD-Net may fail to detect long and thin objects, like the long bridges. This is a common constraint of proposal-based object detectors like Faster R-CNN [21], which can

TABLE 3.3: Ablation study on the design choices of our proposed CAD-Net. Results are reported on the DOTA validation set.

Design Choices	**mAP (%)**
Baseline (Faster R-CNN with FPN)	59.8
Baseline + GCNet	62.4
Baseline + PLCNet	61.2
Baseline + Spatial-and-Scale-Aware Attention	62.4
Baseline + GCNet + PLCNet	62.9
Baseline + GCNet + Spatial-and-Scale-Aware Attention	63.8
Baseline + PLCNet + Spatial-and-Scale-Aware Attention	63.9
CAD-Net (Ours)	64.8

only employ a limited number of anchors for objects with limited aspect ratios. Fourth, CAD-Net still tends to miss detecting objects that are heavily overlapped with each other. We believe this issue can be better addressed by setting appropriate hyper-parameters for Non-Maximum-Suppression (NMS).

3.3.5 Ablation Study

We conduct ablation experiments to identify the contributions of the proposed GCNet, PLCNet, and spatial-and-scale-aware attention. Note that the ground truth annotations of the test set of DOTA dataset are not publicly accessible, and the number of submissions for evaluation on the test set is also limited by the dataset creators. Therefore, all experiments of this ablation study are evaluated on the DOTA validation set that provides publicly accessible object annotations.

Several models are evaluated for the ablation study, including: **Baseline:** The network in beige in Fig. 3.2, which is actually a Faster R-CNN [21] with FPN [22] using ResNet-101 [76] backbone (we follow all configurations as [21, 22]); **Baseline + GCNet:** The Baseline with GCNet included only; **Baseline + PLCNet:** The Baseline with PLCNet included only, where PLCNet is built upon FPN-generated feature pyramid, as attention modulated feature pyramid does not exist under this setup; **Baseline + Spatial-Scale-Aware Attention:** The Baseline with spatial-and-scale-aware attention included only; and **CAD-Net:** The full implementation of the proposed context-aware detection network as shown in Fig. 3.2. Experiments for paired components are also included to verify the complementarity of the proposed components.

As Table 3.3 shows, the model **Baseline** can only achieve a mAP of 59.8%. By including the GCNet, the model **Baseline + GCNet** achieves a ∼2.5% mAP improvement, demonstrating the effectiveness of including global scene-level contextual information. The model **Baseline + PLCNet** also achieves a ∼1.5% mAP improvement while the local contextual information is included, demonstrating the effectiveness of including neighboring objects/features in object detection in aerialFbold imagery. The model **Baseline + Spatial-Scale-Aware Attention** achieves a similar ∼2.5% mAP improvement when the proposed attention module is included, demonstrating the effectiveness of including our proposed attention module to generate spatial-and-scale-aware feature maps. In addition, three experiments with different paired components are included to demonstrate that the proposed GCNet, PLCNet, and attention module are complementary to each other. Finally, the proposed CAD-Net that combines the GCNet, PLCNet and attention module achieves a 5.0% mAP improvement compared to the **Baseline** model, pushing the mAP to 64.8%. The results of this ablation study align well with our motivations.

3.4 Conclusion

This chapter presents CAD-Net, an accurate and robust context-aware detection network for objects in optical remote sensing images. CAD-Net is tailored for the constrained scenarios of object detection in remote sensing imagery, where there exists severe information loss, lack of visual clue, huge scale variation, densely distributed objects, and arbitrarily oriented objects. To tackle these issues, Global Context Network (GCNet) and Pyramid Local Context Network (PLCNet) are proposed, which extract scene-level and object-level contextual information that is highly correlated to objects of interest to provide extra guidance for object detection in remote sensing images. In addition, a spatial-and-scale-aware attention module is designed, which guides the network to focus on scale-adaptive features and also emphasizes the degraded texture details. Extensive experiments over two publicly available datasets verify the uniqueness of object detection in remote sensing images and also show that the proposed CAD-Net achieves superior object detection performance compared with state-of-the-art generic object detection methods.

This chapter also indicates that generic object detectors, when applied to special scenarios, often require careful and appropriate modifications to tackle those unique challenges within the scenarios to avoid severe performance degradation.

Chapter 4

SAM-DETR++: Accelerating DETR's Convergence via Semantic-Aligned Matching [1] [2]

4.1 Introduction

Object detection [4] is a fundamental computer vision task and has experienced remarkable progress with the recent development of deep learning and convolutional neural networks (ConvNets). However, most modern ConvNet-based object detectors (*e.g.*, Faster R-CNN [21], YOLO [28, 29], FCOS [35]) still heavily rely on a series of hand-crafted components, such as anchors, non-maximum suppression (NMS), rule-based training target assignment, *etc.*, which lead to complex detection pipelines and sub-optimal performance. Recently, the emergence of DEtection TRansformer (DETR) [1] has revolutionized the paradigm for object detection. DETR adopts a simple Transformer encoder-decoder pipeline [36] and removes the need for those hand-crafted components, achieving a fully end-to-end framework for object detection. However, despite its simplicity and promising performance, DETR suffers from severely slow convergence on training, requiring 500 epochs to

[1] Part of the work in this chapter has been published as Gongjie Zhang, Zhipeng Luo, Yingchen Yu, Kaiwen Cui, and Shijian Lu. "Accelerating DETR Convergence via Semantic-Aligned Matching." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (**CVPR**), pp. 949-958. 2022. [42]

[2] Part of the work in this chapter has been submitted to and is currently under review at International Journal of Computer Vision (**IJCV**).

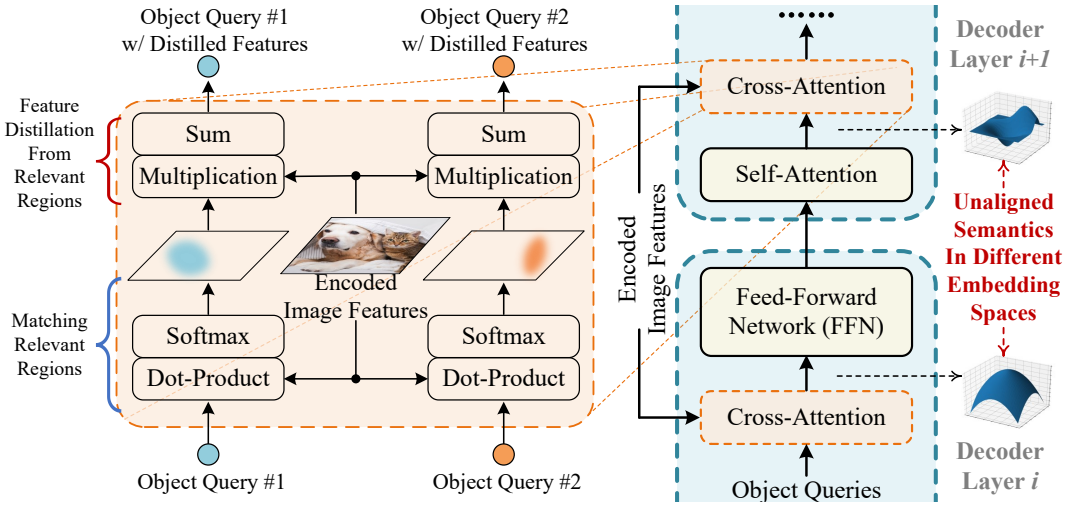


FIGURE 4.1: The analysis for the root of DETR's slow convergence. **Left:** The cross-attention module in DETR's decoder layers can be interpreted as a 'matching and feature distillation' process. Each object query first matches its particular relevant regions in encoded image features via 'Dot-Product and Softmax', and then distills instance-level features from the matched regions for subsequent prediction. **Right:** However, modules between cross-attentions may project object queries and encoded image features into different feature embedding spaces, leading to the unaligned semantics between them. Such unaligned semantics imposes difficulty in cross-attention's matching process and thus hinders the convergence of DETR-based object detection frameworks.

fully converge on the MS COCO dataset [6], while most other ConvNet-based object detectors [21, 22, 28, 29, 32, 35] only requires 12∼36 training epochs instead. DETR's slow convergence significantly increases its training cost and thus hinders the wide application of DETR.

DETR uses a set of object queries in its decoder to represent potential objects at different spatial locations. As shown in Fig. 4.1 (left), in the cross-attention modules of DETR's decoder layers, these object queries interact with the encoded image features through a 'matching and feature distillation' process, where each object query first matches its relevant regions in the encoded image features, and then distills corresponding instance-level features from the matched regions. The object queries after distilling relevant features are used to generate instance-level detection predictions as well as to repeat the subsequent 'matching and feature distillation' processes for refined predictions. However, as pointed out in [38–40, 84], it is difficult for object queries to learn to match appropriate regions. As illustrated in Fig. 4.1 (right), we observe that the matching difficulty is largely attributed to

the unaligned semantics between object queries and encoded image features. Concretely, the modules between cross-attentions project object queries into different feature embedding spaces, in which object queries have different feature semantics from encoded image features. This leads to the complication of matching object queries with relevant regions and further DETR's slow convergence.

A few recent works have been proposed to tackle the slow convergence issue of DETR. For example, Deformable DETR [38] replaces the original dense attention with sparse deformable attention; Conditional DETR [39] and SMCA-DETR [84] propose conditioned cross-attention and spatially modulated co-attention (SMCA), respectively, to replace the cross-attention module in DETR's decoder, aiming to impose spatial constraints to the original cross-attention to better focus on prominent regions; the recently proposed DN-DETR [85] designs a novel de-noising training strategy to speed up DETR's training procedure, which also achieves very promising results. In contrast, in this chapter, we aim to tackle the slow convergence issue of DETR from a different perspective without modifying DETR's original attention mechanism or training strategy.

Our core idea is to ease the matching process between object queries and their corresponding target features. An intuitive and promising direction to mitigate the matching difficulty caused by unaligned semantics has been explored in Siamese-based architectures, which adopt identical sub-networks to produce comparable output feature vectors for similarity computation. The effectiveness of Siamese-based architectures has been extensively verified in various matching-involved vision tasks, such as object tracking [86–91], re-identification [92–96], and few-shot recognition [43, 97–100]. In light of the success of Siamese-based architectures in matching-involved tasks, we follow the similar philosophy to address the matching complication in the cross-attention module of DETR's decoder.

With these motivations, in this chapter, we propose *Semantic-Aligned-Matching DETR (SAM-DETR)* that accelerates the convergence of DETR via a semantic-aligned matching mechanism. Concretely, the proposed SAM-DETR appends a plug-and-play module ahead of the cross-attention modules in DETR's decoder layers, with which object queries and encoded image features can be projected into the same semantics-aligned feature embedding spaces and thus be matched efficiently. The aligned semantics imposes a strong prior for each object query to focus on those semantically similar regions in encoded image features. In addition,

motivated by the importance of objects' keypoints and extremities in recognition and localization [1, 39, 101], SAM-DETR also explicitly identifies multiple representative keypoints for each object query and uses their features for semantic-aligned matching, which can naturally fit into the original multi-head attention mechanism [36] for enhanced representation capacity.

Furthermore, the semantic-aligned matching mechanism can be extended to fuse multi-scale features that are inherently unaligned in feature semantics. With the multi-scale feature fusion via semantic-aligned matching, we extend SAM-DETR to SAM-DETR++, which can represent objects at different scales in a 'divide and conquer' manner and significantly alleviate the object representation complexity, yielding even faster convergence and improved accuracy.

Finally, since SAM-DETR++ only introduces a plug-and-play module into the original DETR while leaving most other operations unchanged, it can be easily integrated with existing convergence solutions [84, 85] in a complementary manner, boosting detection performance to a greater extent.

In summary, the contributions of this chapter are summarized below. *First*, we propose *SAM-DETR*, which accelerates DETR's convergence with a plug-and-play module that enables semantic-aligned matching between object queries and encoded image features. *Second*, we propose to explicitly search for objects' representative keypoints and leverage their features for semantic-aligned matching, which further strengthens the representation capacity of the introduced semantic-aligned matching mechanism. *Third*, we further propose *SAM-DETR++*, in which multi-scale feature fusion is incorporated via the semantic-aligned matching mechanism, enabling adaptive representation of objects at different scales and achieving faster convergence as well as superior detection performance. *Fourth*, experiments validate that our proposed SAM-DETR and SAM-DETR++ can achieve significantly faster convergence and higher accuracy than the original DETR [1]. *Fifth*, our approach offers a unique perspective in mitigating DETR's slow convergence issue with simply a plug-and-play module, thus can be easily integrated with existing convergence solution for DETR in a complementary manner. Experiments show that with just 12 training epochs, our fully-fledged SAM-DETR++ surpasses the original DETR [1] trained for 500 epochs on the MS COCO benchmark [6], and achieves the state-of-the-art performance among DETR-based detectors.

4.2 Methodology

In this section, we first review the basic architecture of DETR, and then introduce the architecture of our proposed *Semantic-Aligned-Matching DETR (SAM-DETR)*. We further introduce *SAM-DETR++*, which extends the semantic-aligned matching mechanism for multi-scale feature fusion. We also show how to integrate our methods with existing convergence solutions to boost DETR's convergence and accuracy to a greater extent.

4.2.1 Background and Motivation

Since our proposed method is developed on top of DETR [1] for its accelerated convergence, we first briefly review the basic architecture of DETR [1] before introducing our method.

Unlike ConvNet-based object detectors [21, 29, 31, 34, 35] that address object detection by solving surrogate classification and regression tasks, DETR [1] directly formulates object detection as a set prediction problem. The pipeline of DETR is simple: a backbone network, a Transformer encoder, and a Transformer decoder. Given an input image $\mathbf{I} \in \mathbb{R}^{H_0 \times W_0 \times 3}$, the backbone network and the Transformer encoder produce the encoded features for the input image $\mathbf{F} \in \mathbb{R}^{HW \times d}$, in which d denotes the number of feature channels, and H_0, W_0 and H, W denote the spatial sizes of the image and the encoded features, respectively. After that, the Transformer decoder takes the encoded image features $\mathbf{F}$ and a small set of object queries $\mathbf{Q} \in \mathbb{R}^{N \times d}$ as input, and then produces the detection results. Here, N denotes the number of object queries, which is typically set to 300 [38–41, 43, 84].

The Transformer decoder consists of multiple decoder layers, in which object queries are sequentially processed by a self-attention module, a cross-attention module, and a feed-forward network (FFN) to produce the outputs. The object queries output by each decoder layer are further fed into the subsequent layers and go through a Multi-Layer Perceptron (MLP) to produce detection predictions. The cross-attention module is the key element in the Transformer decoder, in which object queries interact with the encoded image features. As discussed in Section 4.1 and illustrated in Fig. 4.1, the cross-attention module can be interpreted as a 'match and feature distillation' process: object queries first search for the relevant regions

to match, then distill instance-level features from the matched regions to generate detection predictions. We formulate and interpret the cross-attention mechanism as:

$$\mathbf{Q}' = \overbrace{\underbrace{\text{Softmax}(\frac{(\mathbf{Q}\mathbf{W}_\text{q})(\mathbf{F}\mathbf{W}_\text{k})^\text{T}}{\sqrt{d}})}_{\text{to distill features from relevant regions}}(\mathbf{F}\mathbf{W}_\text{v})}^{\text{to match relevant regions}}, \tag{4.1}$$

where $\mathbf{W}_\text{q}$, $\mathbf{W}_\text{k}$, and $\mathbf{W}_\text{v}$ denote the linear projections for query, key, and value, respectively, in the Transformer attention mechanism, and $\mathbf{Q}' \in \mathbb{R}^{N \times d}$ denotes the cross-attention's generated object queries.

Preferably, the cross-attention's output object queries $\mathbf{Q}'$ should contain instance-level features distilled from the corresponding regions, which are used to produce detection predictions. However, as discussed before and also verified in [38–41, 84], the object queries are initially equally matched to all spatial locations in the encoded image features and are very challenging to learn to focus on specific regions properly. The matching complication with unaligned semantics is the key root for DETR's slow convergence.

4.2.2 SAM-DETR

Our proposed SAM-DETR aims to relieve the difficulty of the matching process in Eq. 4.1 by semantically aligning object queries and encoded image features into the same embedding space, thus accelerating DETR's convergence. Its major difference from the original DETR [1] lies in the Transformer decoder layers. As illustrated in Fig. 4.2 (a), the proposed SAM-DETR appends a *Semantics Aligner* module ahead of the cross-attention module and models learnable *reference boxes* to facilitate the matching process. Same as DETR [1], the decoder layer is repeated six times, with zeros as input for the first layer and previous layer's outputs as input for subsequent layers.

The learnable reference boxes $\mathbf{R}_\text{box} \in \mathbb{R}^{N \times 4}$ are modeled at the first decoder layer, representing the initial locations of the corresponding object queries. With the localization guidance of these reference boxes, the proposed Semantics Aligner takes the preceding object query embeddings $\mathbf{Q}$ and the encoded image features $\mathbf{F}$ as inputs to generate new object query embeddings $\mathbf{Q}^\text{new}$ and their position

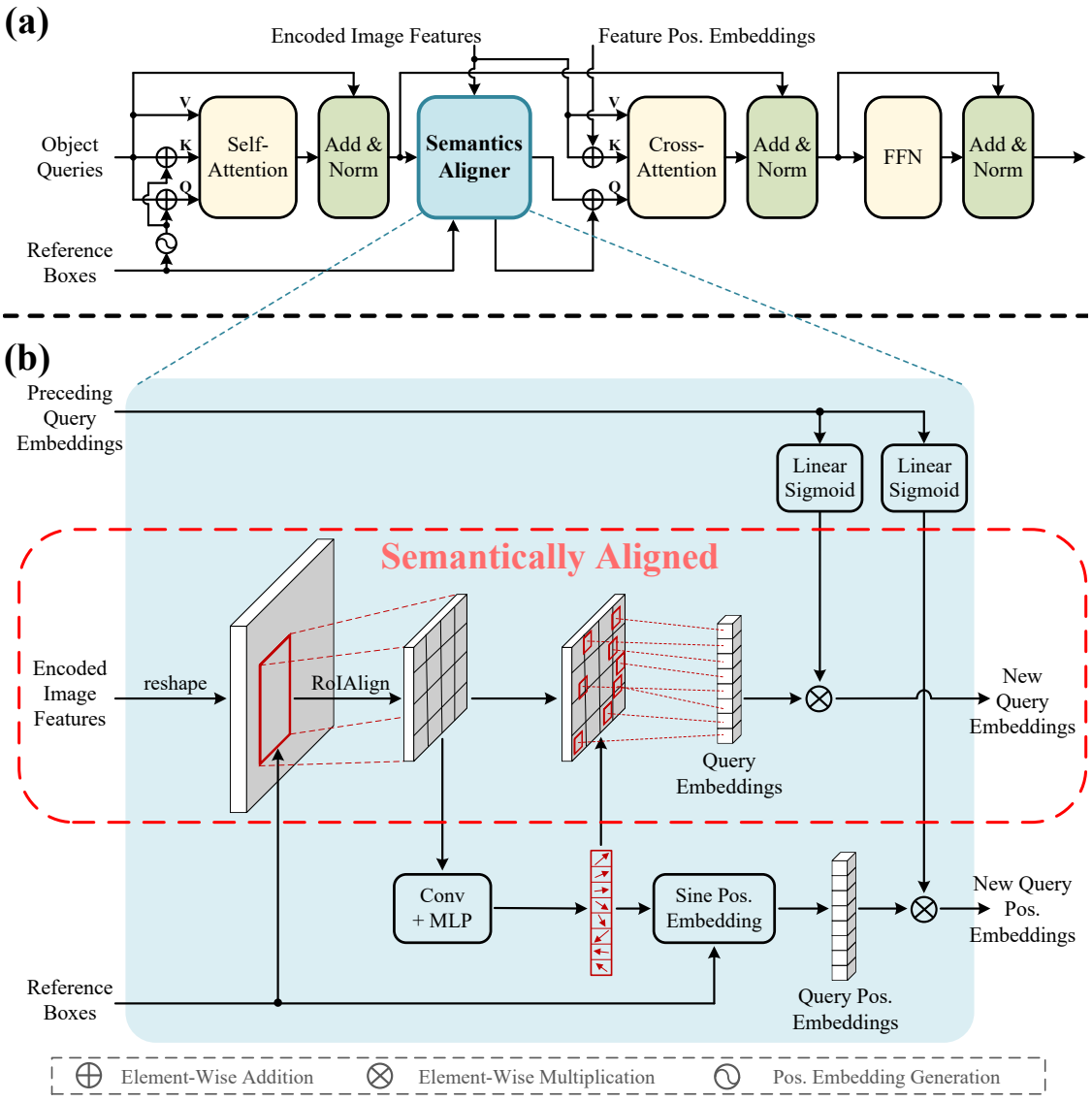


FIGURE 4.2: The proposed SAM-DETR appends a Semantics Aligner into the Transformer decoder layer. **(a) The architecture of one decoder layer in SAM-DETR.** It models a learnable reference box for each object query, whose center location is used to generate corresponding position embeddings. With the guidance of the reference boxes, Semantics Aligner generates new object queries that are semantically aligned with the encoded image features, thus facilitating their subsequent matching. **(b) The pipeline of the proposed Semantics Aligner.** For simplicity, only one object query is illustrated. It first leverages the reference box to extract features from the corresponding region via RoIAlign. The region features are used to predict the coordinates of representative keypoints with the most discriminative features. The representative keypoints' features are then extracted as the new query embeddings with aligned semantics, which are further reweighted by preceding query embeddings to incorporate useful information from them.

embeddings $\mathbf{Q}_{\text{pos}}^{\text{new}}$, feeding to the subsequent cross-attention module. The generated embeddings $\mathbf{Q}^{\text{new}}$ are enforced to lie in the same embedding space with the encoded image features $\mathbf{F}$, which facilitates the subsequent matching process between them, making object queries able to quickly and properly attend to relevant regions in the encoded image features.

4.2.2.1 Semantics Aligner for Semantic-Aligned Matching

As formulated in Eq. 4.1 and illustrated in Fig. 4.1 (left), the cross-attention module uses dot-product to produce the attention weight maps, which represent the matching between object queries and encoded image features. It is natural and intuitive to adopt dot-product for generating the attention weight maps, as dot-product is a good metric for the similarity between two feature vectors, which encourages object queries to have higher attention weights for regions with higher similarities. However, as illustrated in Fig. 4.1 (right), the modules between cross-attentions perform projections on object queries, which causes object queries and encoded image features to be projected into different feature embedding spaces, leading to the unaligned semantics between them. The unaligned semantics causes each object query to almost equally match all spatial locations within the encoded image features at initialization, adding substantial complexity for learning a meaningful matching between them.

Motivated by the above observations, we design Semantics Aligner to ensure that object query embeddings are within the same feature embedding space as encoded image features before being processed by cross-attention. This guarantees that the dot-product between query embeddings and encoded image features is always a meaningful measurement of similarity without the need to be explicitly learned, which imposes a prior for object queries to match relevant regions with similar semantics and reduces the matching complication.

The semantics alignment is achieved by re-sampling new object queries from the encoded image features. Concretely, as shown in Fig. 4.2 (b), Semantics Aligner first restores the encoded image features' spatial dimensions from 1D sequences $HW \times d$ to 2D maps $H \times W \times d$. Then, Semantics Aligner extracts regional features $\mathbf{F}_{\text{R}} \in \mathbb{R}^{N \times 7 \times 7 \times d}$ from the encoded image features via RoIAlign [23] from the corresponding reference boxes. Finally, the new object query embeddings $\mathbf{Q}^{\text{new}}$

and new query positional embeddings $\mathbf{Q}_{\text{pos}}^{\text{new}}$ are obtained by re-sampling features from $\mathbf{F}_{\text{R}}$. Mathematically, our proposed Semantics Aligner can be formulated at a high level as:

$$\mathbf{F}_{\text{R}} = \text{RoIAlign}(\mathbf{F}, \mathbf{R}_{\text{box}}), \tag{4.2}$$

$$\mathbf{Q}^{\text{new}}, \mathbf{Q}_{\text{pos}}^{\text{new}} = \text{Re-Sample}(\mathbf{F}_{\text{R}}, \mathbf{R}_{\text{box}}, \mathbf{Q}). \tag{4.3}$$

As the re-sampling procedure does not involve any projection (*e.g.*, ConvNet or MLP), the new object query embeddings $\mathbf{Q}^{\text{new}}$ always lie within the same feature embedding space as the encoded image features $\mathbf{F}$, which encourages object queries to focus on semantically similar regions in the following cross-attention module. The design choice for the re-sampling procedure is to be detailed later.

4.2.2.2 Matching with Representative Keypoint Features

The re-sampling process can be easily accomplished by simple operations like applying global average pooling or global max pooling on region features $\mathbf{F}_{\text{R}}$. But instead, we propose a more sophisticated approach to re-sample new object query embeddings, inspired by the prior works [1, 39, 101–103] that identify the importance of objects' representative keypoints in object detection. Specifically, Semantics Aligner explicitly searches for multiple representative keypoints for each object query and extracts their features for the aforementioned semantic-aligned matching. Such design can naturally fit in the multi-head attention mechanism [36] without any modification, enabling every attention head to produce different weights to focus on different parts.

Here, we denote the number of attention heads as M, which is set to 8 in most DETR-based object detectors [1, 38–41, 84]. M is also the number of representative keypoints to search for each object query. As shown in Fig. 4.2 (b), after retrieving region features $\mathbf{F}_{\text{R}}$, Semantics Aligner adopts a ConvNet followed by an MLP to predict the spatial locations of the M keypoints for each object query, representing the keypoints that are crucial for recognizing and localizing the potential objects, which can be formuated as:

$$\mathbf{R}_{\text{SP}} = \text{MLP}(\text{ConvNet}(\mathbf{F}_{\text{R}})), \tag{4.4}$$

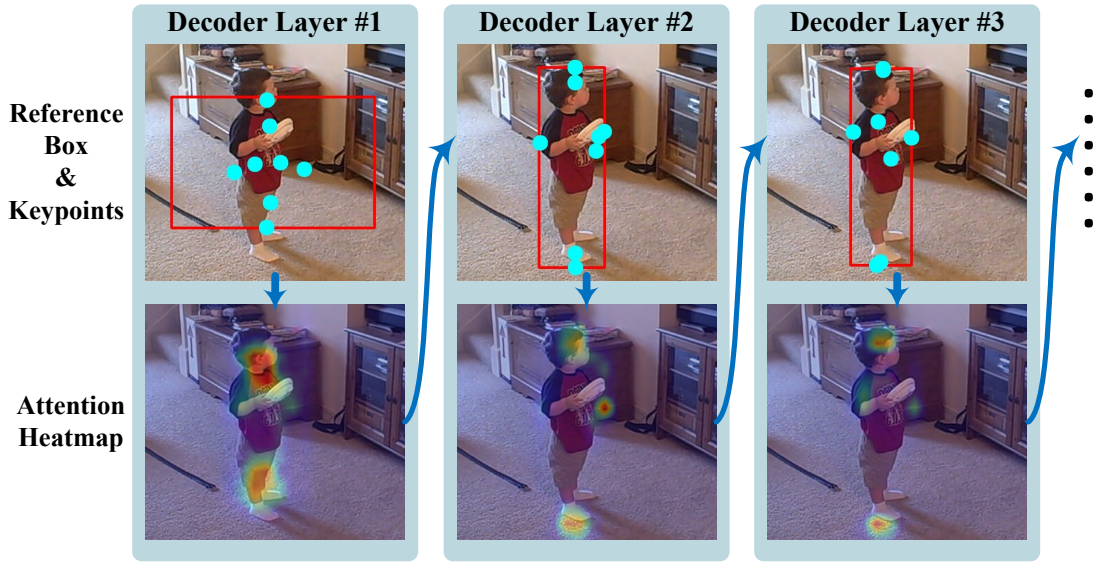


FIGURE 4.3: The proposed Semantics Aligner in each decoder layer searches multiple representative keypoints (cyan dots) within each reference box (red box), and uses their features for the subsequent semantic-aligned matching. As detection proceeds, the keypoints gradually fall on salient and semantically meaningful locations, and the attention heatmaps gradually become more precise and focused.

where $\mathbf{R}_{\mathrm{SP}} \in \mathbb{R}^{N \times M \times 2}$ denotes the locations of M keypoints for N object queries. Note that the predicted coordinates are constrained to be inside their corresponding reference boxes. This design choice has been empirically verified in Section 4.3.5. With the predicted $\mathbf{R}_{\mathrm{SP}}$, the features of these representative keypoints can be then sampled from $\mathbf{F}_{\mathrm{R}}$ via bi-linear interpolation. Semantics Aligner finally concatenates the M sampled features vectors corresponding to the M representative keypoints as the new query embeddings, which are fed into the multi-head cross-attention so that each attention head can focus on features of one representative keypoint. Similarly, the object queries' corresponding positional embeddings can be computed using sinusoidal functions based on the keypoints' image-scale coordinates, and then are also concatenated to feed into the multi-head cross-attention module.

$$\mathbf{Q}^{\mathrm{new}\prime} = \mathrm{Concat}(\{\mathbf{F}_{\mathrm{R}}[..., x, y, ...] \text{ for } x, y \in \mathbf{R}_{\mathrm{SP}}\}) \tag{4.5}$$

$$\mathbf{Q}^{\mathrm{new}\prime}_{\mathrm{pos}} = \mathrm{Concat}(\mathrm{Sinusoidal}(\mathbf{R}_{\mathrm{box}}, \mathbf{R}_{\mathrm{SP}})) \tag{4.6}$$

Fig. 4.3 visualizes the searched representative keypoints and the attention heatmaps

produced by semantic-aligned matching. It can be observed that the search keypoints gradually fall on the salient locations with rich semantics (*e.g.*, head and extremities) as detection proceeds. In addition, the attention heatmaps also gradually become more precise and focus on those semantically meaningful regions. These results validate the effectiveness of searching and exploiting keypoint features for semantic-aligned matching in enhancing the representation capacity.

4.2.2.3 Feature Reweighting by Preceding Query Embeddings

So far, Semantics Aligner can generate new object query embeddings $\mathbf{Q}^{\text{new}\prime}$ with aligned semantics with the encoded image features. However, it also brings one issue: the cross-attention cannot leverage the preceding query embeddings $\mathbf{Q}$ that contain valuable information for detection. To mitigate this issue, Semantics Aligner further receives the preceding query emebddings $\mathbf{Q}$ as inputs to produce a set of reweighting coefficients via 'Linear Projection + Sigmoid'. The reweighting coefficients are applied to new query embeddings $\mathbf{Q}^{\text{new}\prime}$ and their positional embeddings $\mathbf{Q}_{\text{pos}}^{\text{new}\prime}$ through element-wise multiplication, highlighting those important features. As a result, the useful information from preceding query embeddings can be effectively leveraged in cross-attention for the semantic-aligned matching. Note that feature reweighting does not affect the aligned semantics of query embeddings, as it does not perform any projection on the query embeddings. This feature reweighting process can be formulated as:

$$\mathbf{Q}^{\text{new}} = \mathbf{Q}^{\text{new}\prime} \otimes \sigma(\mathbf{Q}\mathbf{W}_{\text{RW1}}), \tag{4.7}$$

$$\mathbf{Q}_{\text{pos}}^{\text{new}} = \mathbf{Q}_{\text{pos}}^{\text{new}\prime} \otimes \sigma(\mathbf{Q}\mathbf{W}_{\text{RW2}}), \tag{4.8}$$

where $\mathbf{W}_{\text{RW1}}$ and $\mathbf{W}_{\text{RW2}}$ are the learnable parameters for linear projections, $\sigma(\cdot)$ denotes sigmoid function, and $\otimes$ denotes element-wise multiplication.

4.2.3 SAM-DETR++

On top of the SAM-DETR introduced above, we further design SAM-DETR++ with two minor modifications incorporated. These two simple modifications further improve the convergence speed and detection performance.

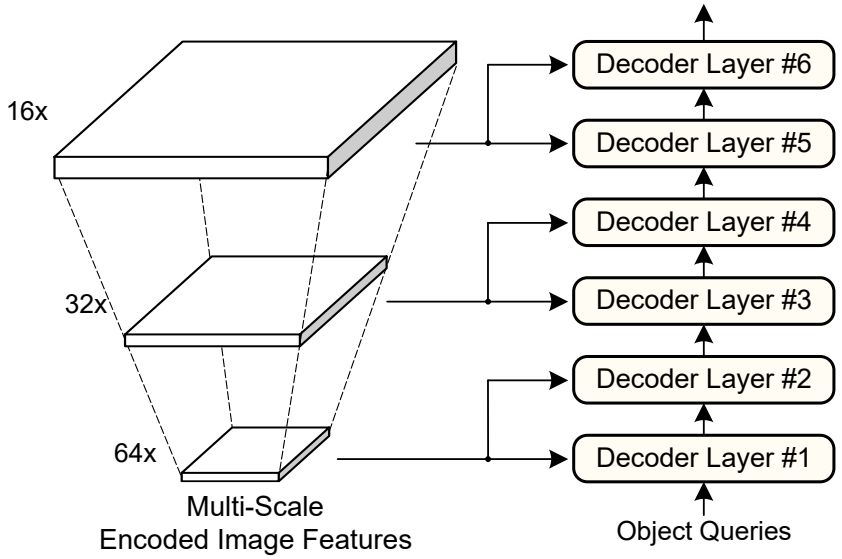


FIGURE 4.4: The proposed SAM-DETR++ can fuse multi-scale features by simply feeding different feature scales into different decoder layers in a coarse-to-fine manner. Thanks to the introduced semantic-aligned matching mechanism, SAM-DETR++ can effectively fuse multi-scale features that are inherently unaligned in feature semantics.

4.2.3.1 Multi-scale Feature Fusion with Aligned Semantics

Detecting objects of vastly different scales has always been one major challenge in object detection. Modern ConvNet-based object detectors (*e.g.,* Faster R-CNN [21] w/ FPN [22], M2Det [104], EfficientDet [37]) usually incorporate multi-scale features to accommodate this issue by representing objects at different scales in a 'divide and conquer' manner, which reduces the representation complexity and achieves superior detection accuracy and faster convergence. With this motivation, we further extend the proposed semantic-aligned matching strategy to fuse multi-scale features in a coarse-to-fine manner.

As shown in Fig. 4.4, the proposed method to fuse multi-scale features is simple and concise. Considering the cascade nature of DETR's decoder, we feed the features of different scales into different stages of the decoder, making the detection pipeline a coarse-to-fine refinement process. Concretely, the first two decoder layers receive the coarsest feature maps to reduce search space for initial localization; the subsequent two layers receive finer feature maps for more precise localization; the last two layers receive high-resolution feature maps for detecting tiny objects. This

simple and effective design does not introduce extra parameters but allows SAM-DETR++ to adaptively fuse multi-scale features to represent objects at different scales, thus greatly lowering the learning complexity.

It is worth noting that it is our proposed semantic-aligned matching mechanism that enables this simple approach to fuse multi-scale features effectively. Experiments in Section 4.3.5 show that without the proposed semantic-aligned matching, directly fusing multi-scale features does not bring a clear performance gain. This is because of the inevitable unaligned semantics across different feature scales, which causes extra complexity in the matching processes between object queries and encoded image features, as discussed before. Our introduced Semantics Aligner alleviates this issue by explicitly aligning the semantics between object query embeddings and encoded image features at all decoder layers, thus enabling the effective fuse of features from different scales.

4.2.3.2 Removing Dropout in Transformer

Most existing DETR-like detectors [1, 38, 39, 42, 43, 68, 84] contain dropout [105] in the Transformer encoder-decoder architecture [36]. Dropout [105] is included to mitigate the overfitting issue in natural language processing (NLP) tasks. However, in the task of object detection within images, we empirically find that dropout does not mitigate overfitting but harms the performance of object detection. We conjecture that this is largely attributed to the unique property of object detection within 2D images, where adjacent pixels within feature maps are strongly correlated. Therefore, we incorporate a minor tweak to remove the dropout in the Transformer, which yields better performance at no computational cost. The effectiveness of this modification is verified in Section 4.3.5.

4.2.4 Network Optimization

As described in Section 4.2.2 and Section 4.2.3, the appended operations of SAM-DETR and SAM-DETR++, including the query re-sampling process, the exploitation of features from multiple representative keypoints, and the feature reweighting

process, are all fully differentiable. Therefore, SAM-DETR and SAM-DETR++ require no additional loss function for training, *i.e.*, SAM-DETR and SAM-DETR++ can be learned purely from the supervision signals of the original DETR [1].

4.2.5 Compatibility with Existing Convergence Solutions

As shown in Fig. 4.2 (a), our methods only append a plug-and-play module into the Transformer decoder layer, leaving most other operations unchanged. Besides, our methods speed up DETR's training convergence from a distinct perspective from existing convergence solutions. These properties make it easy and effective to integrate the proposed SAM-DETR and SAM-DETR++ with other approaches to achieve even faster convergence and superior detection accuracy. Here, we integrate our methods with two recent works to validate the strong compatibility of our methods.

4.2.5.1 Compatibility with SMCA-DETR

SMCA-DETR [84] replaces DETR's original cross-attention module with Spatially Modulated Co-Attention (SMCA), which estimates the position of each object query, and then applies a series of 2D-Gaussian weight maps to constrain the attention responses in different attention heads. Both the center locations and the scales for SMCA's 2D-Gaussian weight maps are predicted from the corresponding object query embeddings. SMCA [84] effectively accelerates DETR's convergence by imposing spatial constraints to the SMCA module.

To integrate SMCA [84] with our proposed SAM-DETR and SAM-DETR++, we make one minor modification to the SMCA mechanism: we adopt the coordinates of the M representative keypoints as the central locations for the 2D Gaussian-like weight maps. The scales of the weight maps are also predicted from the region features in parallel to the central locations. Experimental results in Section 4.3.4 validate the complementary effect between our proposed methods and SMCA [84].

4.2.5.2 Compatibility with DN-DETR

The recently proposed DN-DETR [85] introduces a novel de-noising training strategy to speed up DETR's training procedure, which is also complementary to our approach without any adaptation. Experiment results in Section 4.3.4 also validate the complementary effect between our proposed methods and DN-DETR [85].

4.3 Experiments

4.3.1 Dataset and Evaluation Metrics

Following prior works [1, 38–41, 84], we mainly perform the experiments on the MS COCO 2017 dataset [6], using ∼117k images in `train2017` for training and 5k images in `val2017` for evaluation. We adopt MS COCO's standard evaluation metrics for performance evaluation.

4.3.2 Implementation Details

The implementation details of SAM-DETR and SAM-DETR++ mostly align with the original DETR [1] and other prior works [38–41, 84]. We use ImageNet-pretrained [46] ResNet-50 [76] as the backbone network. All experiments are performed on servers with $8 \times$ NVIDIA V100 GPUs. We train our models with AdamW optimizer [106, 107]. The batch size is set to 16 for training, except when ResNet-50-DC5 is used as the backbone, the batch size is set to 8. The initial learning rate is 1×10^{-5} for the backbone parameters and 1×10^{-4} for the other parameters. The weight decay is set to 1×10^{-4}. Two training schedules are experimented: *(i)* the 12-epoch (1x) schedule that is widely adopted in ConvNet-based detectors [21, 22, 32, 35], where the learning rate decays at the 10^{th} epoch; *(ii)* the 50-epoch schedule that is often used in Transformer-based detectors [38–41, 84], where the learning rate decays at the 40^{th} epoch. Model-related hyper-parameters (*e.g.*, feature channel dimension, number of encoder and decoder layers) remain the same with DETR [1], except we make two minor modifications following some recent works [38–40, 84] to improve DETR's convergence speed: the number of object queries N is increased from 100 to 300; the sigmoid focal loss [32] is adopted

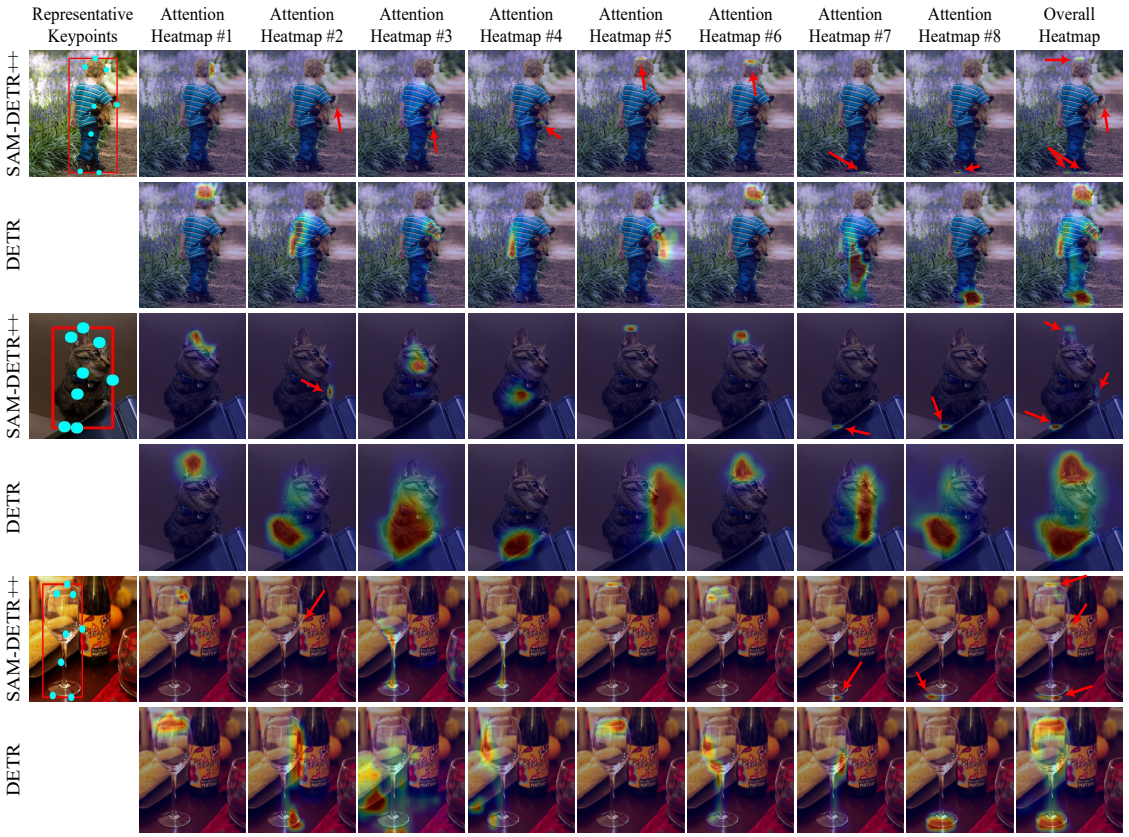


FIGURE 4.5: Visualization of the searched representative keypoints and the attention heatmaps of different attention heads in cross-attention from our proposed SAM-DETR++. The searched representative keypoints mostly fall around objects of interest, and typically fall on the positions with the most distinctive features for recognition or localization, like object extremities or central points. Our method's attention heatmaps are much more focused compared with the original DETR without semantic-aligned matching, which proves the effectiveness of our approach in relieving the complication in the matching processes between object queries and encoded image features, which accelerates DETR's convergence. Red-colored arrows highlight those fine details in attention heatmaps. Zoom-in may be required to view details.

as the classification loss instead of the cross-entropy loss. These two modifications are also applied to the original DETR [1] for a fair comparison with the baseline.

The same data augmentation scheme used in prior works [1, 38–41] is adopted, which includes random resize, horizontal flip, and random crop. We constrain the training images' longest sides to be less or equal than 1333 pixels and the shortest sides to be larger or equal than 480 pixels.

4.3.3 Visualization and Analysis

Fig. 4.5 visualizes the representative keypoints searched by the proposed Semantics Aligner as well as their corresponding attention heatmaps generated from the subsequent multi-head cross-attention module. We also compare the attention heatmaps with the ones generated from the original DETR [1]. Results are obtained under the 12-epoch (1x) training schedule using ResNet-50 [76] as the backbone.

The visualization shows that the searched representative keypoints mostly fall around the target objects, and typically at those representative positions with the most distinctive features, such as object extremities or central points. The attention response heatmaps generated by the subsequent cross-attention modules also show high responses on those searched representative keypoints accordingly. In addition, compared with the original DETR [1], our method shows clearly more precise and focused responses, which validates that the proposed semantic-aligned matching mechanism successfully facilitates the matching of object queries with appropriate regions for distillation of relevant instance-level features, thus accelerating DETR's convergence.

4.3.4 Experiment Results

Results under the 12-epoch (1x) Schedule. Table 4.1 presents object detection results trained with the standard 12-epoch (1x) learning scheme. The table shows that DETR [1] is severely under-trained within 12 epochs due to its slow convergence, while conventional detectors like Faster R-CNN [21] can achieve relatively satisfactory performance. Several recent works [38, 39, 84] modify the original attention mechanism and effectively boost DETR's performance under the 12-epoch training scheme, but still have large gaps compared with the conventional detectors. As a standalone method, our proposed SAM-DETR can achieve a significant performance gain compared with the original DETR baseline (+10.8% AP w/ ResNet-50 and +12.4% AP w/ ResNet-50-DC5), surpassing all DETR-like detectors. Furthermore, being like a plug and play, the proposed SAM-DETR can be easily integrated with existing convergence-boosting solution [84] for DETR to achieve even better performance. Combining our proposed SAM-DETR with spatially modulated co-attention (SMCA) [84] brings consistent improvement over SMCA-DETR [84] and

TABLE 4.1: Object detection performance under the 12-epoch (1x) training schedule on MS COCO val 2017. "‡" denotes the original DETR baseline with increased number of object query (100→300) and focal loss as the classification loss function.

Method	#Epochs	#Params	FLOPs	AP	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L
• ***Backbone: ResNet-50***									
Faster-RCNN-R50 [21]	12	34 M	547 G	35.7	56.1	**38.0**	**19.2**	**40.9**	48.7
DETR-R50 [1] ‡	12	41 M	86 G	22.3	39.5	22.2	6.6	22.8	36.6
Deformable-DETR-R50 (Single-Scale) [38]	12	34 M	78 G	31.8	51.4	33.5	15.0	35.7	44.7
Conditional-DETR-R50 [39]	12	44 M	90 G	32.2	52.1	33.4	13.9	34.5	48.7
SMCA-DETR-R50 (Single-Scale) [84]	12	42 M	86 G	31.6	51.7	33.1	14.1	34.4	46.5
SAM-DETR-R50 (Ours)	12	58 M	100 G	33.1	54.2	33.7	13.9	36.5	51.7
SAM-DETR-R50 w/ SMCA (Ours)	12	58 M	100 G	**36.0**	**56.8**	37.3	15.8	39.4	**55.3**
• ***Backbone: ResNet-50-DC5***									
Faster-RCNN-R50-DC5 [21]	12	166 M	320 G	37.3	58.8	39.7	20.1	41.7	50.0
DETR-R50-DC5 [1] ‡	12	41 M	187 G	25.9	44.4	26.0	7.9	27.1	41.4
Deformable-DETR-R50-DC5 (Single-Scale) [38]	12	34 M	128 G	34.9	54.3	37.6	19.0	38.9	47.5
Conditional-DETR-R50-DC5 [39]	12	44 M	195 G	35.9	55.8	38.2	17.8	38.8	52.0
SMCA-DETR-R50-DC5 (Single-Scale) [84]	12	42 M	187 G	32.5	52.8	33.9	14.2	35.4	48.1
Anchor-DETR-R50-DC5 [40]	12	39 M	151 G	37.1	57.8	39.1	19.0	40.8	51.4
DAB-DETR-R50-DC5 [41]	12	44 M	216 G	38.0	60.3	39.8	19.2	40.9	55.4
SAM-DETR-R50-DC5 (Ours)	12	58 M	210 G	38.3	59.1	40.1	21.0	41.8	55.2
SAM-DETR-R50-DC5 w/ SMCA (Ours)	12	58 M	210 G	**40.6**	**61.1**	**42.8**	**21.9**	**43.9**	**58.5**
• ***Backbone: ResNet-50 (Multi-Scale Features)***									
Faster-R-CNN-R50-FPN [21, 22]	12	42 M	180 G	37.9	58.8	41.1	22.4	41.1	49.1
Cascade-R-CNN-R50-FPN [22, 24]	12	69 M	230 G	40.4	58.9	44.1	22.8	43.7	54.0
FCOS-R50 [35]	12	32 M	201 G	38.6	57.2	41.7	23.5	42.8	48.9
Sparse-R-CNN-R50-FPN [108]	12	106 M	166 G	40.1	59.4	43.5	22.9	43.6	52.9
Deformable-DETR-R50 [38]	12	40 M	173 G	37.2	55.5	40.5	21.1	40.7	50.5
SMCA-DETR-R50 [84]	12	40 M	152 G	35.0	54.1	37.8	18.7	37.7	48.1
SAM-DETR++-R50 (Ours)	12	55 M	203 G	41.9	60.5	45.3	24.6	45.5	57.4
SAM-DETR++-R50 w/ SMCA (Ours)	12	55 M	203 G	43.2	61.5	46.5	25.5	46.5	58.6
SAM-DETR++-R50 w/ SMCA + DN (Ours)	12	55 M	203 G	**44.8**	**62.6**	**47.9**	**26.7**	**48.2**	**60.9**

the standalone version of the proposed SAM-DETR. Besides, our proposed SAM-DETR w/ SMCA even outperforms fast-converging Faster R-CNN [21] under the 12-epoch (1x) training schedule, especially when using higher-resolution features (ResNet-50-DC5).

Fig. 4.6 further presents the convergence curves of SAM-DETR and other competing methods under the 12-epoch (1x) training schedule, with ResNet-50 and ResNet-50-DC5 as backbones, respectively. As shown, using ResNet-50 as backbones, SAM-DETR converges significantly faster than the original DETR baseline [1], and can work in complementary with existing convergence-boosting solution [84], even surpassing the convergence speed of Faster R-CNN [21]. Our proposed SAM-DETR and SAM-DETR w/ SMCA can benefit more from the stronger ResNet-50-DC5 backbones, outperforming other competing methods by larger margins. The results demonstrate the effectiveness of our proposed semantic-aligned

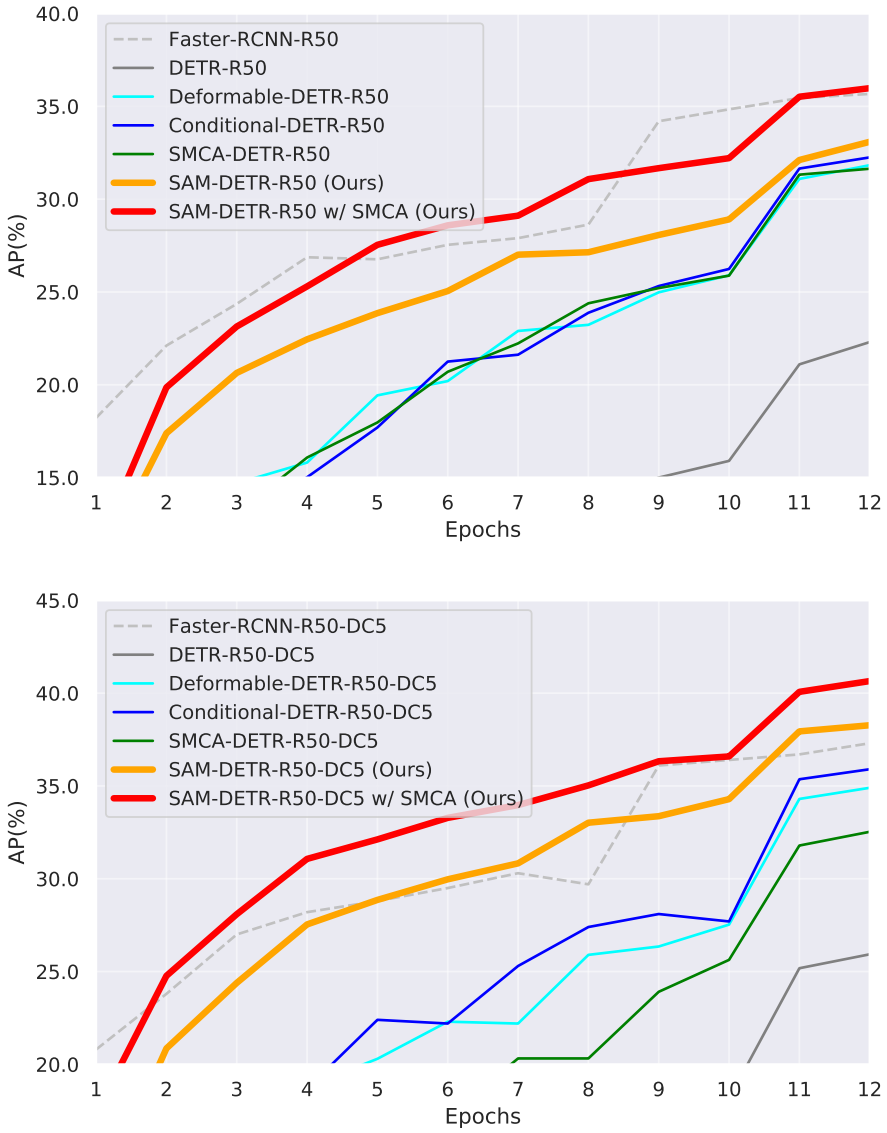


FIGURE 4.6: Convergence curves of SAM-DETR and other detectors on MS COCO val 2017 under the 12-epoch (1x) training scheme with ResNet-50 and ResNet-50-DC5 as backbones. All competing methods are single-scale. SAM-DETR converges much faster than the original DETR, and can work in complementary with existing convergence-boosting solution, surpassing the convergence speed of Faster R-CNN.

matching mechanism in accelerating the convergence speed of DETR.

In Table 4.1, we also present the object detection performance of our proposed SAM-DETR++ that is extended to fuse multi-scale features, as well as the performance of other detectors exploiting multi-scale features. As shown in Table 4.1, fusing multi-scale features via the introduced semantic-aligned matching mechanism significantly improves the detection accuracy over SAM-DETR-R50-DC5

TABLE 4.2: Comparison of SAM-DETR with state-of-the-art object detectors on MS COCO val 2017 under longer training schedules. "‡" denotes the original DETR baseline with increased number of object query (100→300) and focal loss as the classification loss function.

Method	multi-scale	#Epochs	#Params	FLOPs	AP	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L
• ***Baseline methods trained for long epochs:***										
Faster-RCNN-R50-DC5 [21]		108	166 M	320 G	41.1	61.4	44.3	22.9	45.9	55.0
Faster-RCNN-FPN-R50 [21, 22]	✓	108	42 M	180 G	42.0	62.1	45.5	26.6	45.4	53.4
DETR-R50 [1]		500	41 M	86 G	42.0	62.4	44.2	20.5	45.8	61.1
DETR-R50-DC5 [1]		500	41 M	187 G	43.3	63.1	45.9	22.5	47.3	61.1
• ***Comparison of SAM-DETR with other detectors under longer training schemes:***										
Faster-RCNN-R50 [21]		36	34 M	547 G	38.4	58.7	41.3	20.7	42.7	53.1
DETR-R50 [1] ‡		50	41 M	86 G	34.9	55.5	36.0	14.4	37.2	54.5
Deformable-DETR-R50 (Single-Scale) [38]		50	34 M	78 G	39.4	59.6	42.3	20.6	43.0	55.5
Conditional-DETR-R50 [39]		50	44 M	90 G	40.9	61.8	43.3	20.8	44.6	59.2
SMCA-DETR-R50 (Single-Scale) [84]		50	42 M	86 G	41.0	-	-	21.9	44.3	59.1
SAM-DETR-R50 (Ours)		50	58 M	100 G	39.8	61.8	41.6	20.5	43.4	59.6
SAM-DETR-R50 w/ SMCA (Ours)		50	58 M	100 G	41.8	63.2	43.9	22.1	45.9	60.9
Deformable-DETR-R50 [38]	✓	50	40 M	173 G	43.8	62.6	47.7	26.4	47.1	58.0
SMCA-DETR-R50 [84]	✓	50	40 M	152 G	43.7	63.6	47.2	24.2	47.0	60.4
Faster-RCNN-R50-DC5 [21]		36	166 M	320 G	39.0	60.5	42.3	21.4	43.5	52.5
DETR-R50-DC5 [1] ‡		50	41 M	187 G	36.7	57.6	38.2	15.4	39.8	56.3
Deformable-DETR-R50-DC5 (Single-Scale) [38]		50	34 M	128 G	41.5	61.8	44.9	24.1	45.3	56.0
Conditional-DETR-R50-DC5 [39]		50	44 M	195 G	43.8	64.4	46.7	24.0	47.6	60.7
Anchor-DETR-R50-DC5 [40]		50	37 M	172 G	44.2	64.7	47.5	24.7	48.2	60.6
SAM-DETR-R50-DC5 (Ours)		50	58 M	210 G	43.3	64.4	46.2	25.1	46.9	61.0
SAM-DETR-R50-DC5 w/ SMCA (Ours)		50	58 M	210 G	45.0	65.4	47.9	26.2	49.0	63.3
• ***Accelerating DETR's convergence with self-supervised learning:***										
UP-DETR-R50 [109]		150	41 M	86 G	40.5	60.8	42.6	19.0	44.4	60.0
UP-DETR-R50 [109]		300	41 M	86 G	42.8	63.0	45.3	20.8	47.1	61.7

with even reduced computational cost. In addition, it is noteworthy that SAM-DETR++-R50 w/ SMCA+DN achieves a state-of-the-art detection performance of 44.8% AP with only 12 training epochs, which outperforms the original DETR-R50-DC5 [1] trained for 500 epochs (43.3% AP), reducing the required number of training epochs by more than 97.6%. These results show that fusing multi-scale features via our proposed semantic-aligned matching mechanism further improves the detection accuracy and accelerates the convergence to a greater extent.

Results under Longer Training Schedules. Table 4.2 further compares the proposed SAM-DETR with other state-of-the-art object detectors under longer training schedules. Under various setups, the proposed SAM-DETR consistently improves the original DETR's performance, and achieves state-of-the-art accuracy when further integrated with SMCA [84]. The superior performance under various setups demonstrates the effectiveness of the introduced semantic-aligned matching mechanism.

TABLE 4.3: Comparison of SAM-DETR++ with state-of-the-art object detectors on MS COCO val 2017 under longer training schedules. '‡' denotes the original DETR baseline [1] with increased number of object query (100→300) and focal loss as the classification loss function.

Method	#Epochs	#Params	FLOPs	AP	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L
• ***Baseline methods trained for extra-long epochs:***									
Faster-R-CNN-R50-FPN [21, 22]	108	42 M	180 G	42.0	62.1	45.5	26.6	45.4	53.4
DETR-R50 [1]	500	41 M	86 G	42.0	62.4	44.2	20.5	45.8	61.1
DETR-R50-DC5 [1]	500	41 M	187 G	43.3	63.1	45.9	22.5	47.3	61.1
• ***ConvNet-based object detectors:***									
Cascade-Mask-R-CNN-R50-FPN [24]	36	77 M	394 G	44.3	62.2	48.0	26.6	47.7	57.7
TSP-FCOS-R50-FPN [110]	36	52 M	189 G	43.1	62.3	47.0	26.6	46.8	55.9
TSP-R-CNN-R50-FPN [110]	36	64 M	188 G	43.8	63.3	48.3	28.6	46.9	55.7
TSP-R-CNN-R50-FPN [110]	96	64 M	188 G	45.0	64.5	49.6	29.7	47.7	58.0
Sparse-R-CNN-R50-FPN [108]	36	106 M	166 G	45.0	63.4	48.2	26.9	47.2	59.5
• ***Transformer-based object detectors:***									
DETR-R50 [1] ‡	50	41 M	86 G	34.9	55.5	36.0	14.4	37.2	54.5
DETR-R50-DC5 [1] ‡	50	41 M	187 G	36.7	57.6	38.2	15.4	39.8	56.3
UP-DETR-R50 [109]	150	41 M	86 G	40.5	60.8	42.6	19.0	44.4	60.0
UP-DETR-R50 [109]	300	41 M	86 G	42.8	63.0	45.3	20.8	47.1	61.7
Deformable-DETR-R50 [38]	50	40 M	173 G	43.8	62.6	47.7	26.4	47.1	58.0
Deformable-DETR-R50 (two-stage) [38]	50	40 M	173 G	46.2	65.2	50.0	28.8	49.2	61.7
SMCA-DETR-R50 [84]	50	40 M	152 G	43.7	63.6	47.2	24.2	47.0	60.4
SMCA-DETR-R50 [84]	108	40 M	152 G	45.6	65.5	49.1	25.9	49.3	62.6
Conditional-DETR-R50 [39]	50	44 M	90 G	40.9	61.8	43.3	20.8	44.6	59.2
Conditional-DETR-R50 [39]	108	44 M	90 G	43.0	64.0	45.7	22.7	46.7	61.5
Conditional-DETR-R50-DC5 [39]	50	44 M	195 G	43.8	64.4	46.7	24.0	47.6	60.7
Conditional-DETR-R50-DC5 [39]	108	44 M	195 G	45.1	65.4	48.5	25.3	49.0	62.2
Anchor-DETR-R50 [40]	50	37 M	93 G	42.1	63.1	44.9	22.3	46.2	60.0
Anchor-DETR-R50-DC5 [40]	50	37 M	172 G	44.2	64.7	47.5	24.7	48.2	60.6
DAB-DETR-R50 [41]	50	44 M	94 G	42.2	63.1	44.7	21.5	45.7	60.3
DAB-DETR-R50-DC5 [41]	50	44 M	202 G	44.5	65.1	47.7	25.3	48.2	62.3
DAB-DETR-R50-DC5 (3 Patterns) [41]	50	44 M	216 G	45.7	66.2	49.0	26.1	49.4	63.1
Sparse-DETR-R50 [68]	50	41 M	136 G	46.3	66.0	50.1	29.0	49.5	60.8
DN-DETR-R50 [85]	50	44 M	94 G	44.1	64.4	46.7	22.9	48.0	63.4
DN-DETR-R50-DC5 [85]	50	44 M	202 G	46.3	66.4	49.7	26.7	50.0	64.3
SAM-DETR++-R50 (Ours)	50	55 M	203 G	47.5	66.5	51.3	29.3	50.8	62.7
SAM-DETR++-R50 w/ SMCA (Ours)	50	55 M	203 G	48.0	66.6	52.2	29.9	51.5	64.6
SAM-DETR++-R50 w/ SMCA + DN (Ours)	50	55 M	203 G	**49.1**	**67.2**	**53.2**	**30.5**	**52.6**	**64.7**

Table 4.3 further compares SAM-DETR++ (with multi-scale feature fusion) with other state-of-the-art object detectors under the longer training schedules. When trained for 50 epochs, our proposed SAM-DETR++ already outperforms the original DETR [1] trained for 500 epochs by large margins, and also achieves state-of-the-art performance among all Transformer-based object detectors. In addition, as SAM-DETR++ works from a distinct perspective from existing solutions, combining our proposed SAM-DETR++ with SMCA [84] and DN [85] (SAM-DETR++ w/ SMCA and SAM-DETR++ w/ SMCA+DN) brings further performance gains, achieving the state-of-the-art accuracy of 49.1% AP on MS COCO val 2017 with ResNet-50, without bells and whistles. The convergence curves of SAM-DETR++ are presented in Fig. 4.7.

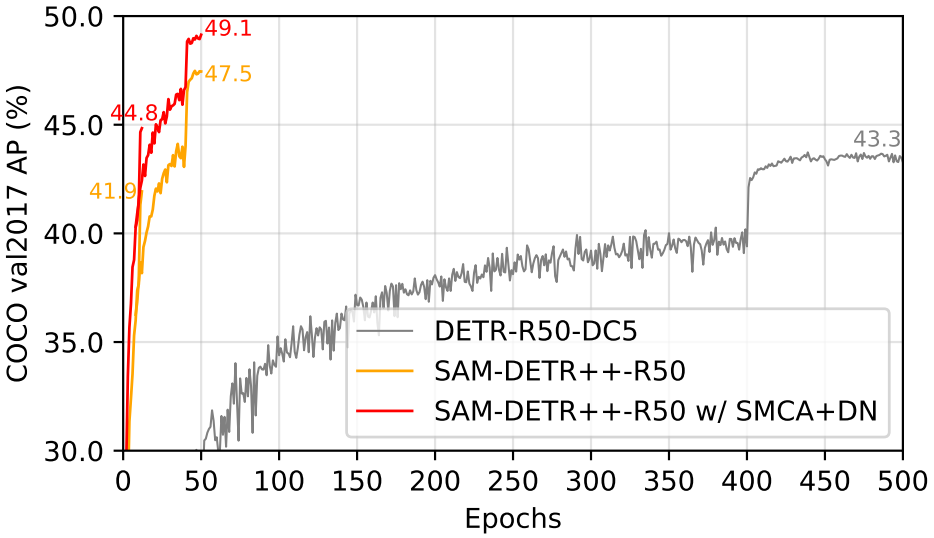


FIGURE 4.7: The convergence curves of DETR and SAM-DETR++. DETR is trained with 500 epochs, with the learning rate dropped at the 400^{th} epoch. SAM-DETR++ are trained under the 12-epoch (1x) and 50-epoch learning schedules. SAM-DETR++ converge much faster and achieve clearly better detection performance over the original DETR.

4.3.5 Ablation Study

We conduct ablation studies to validate the effectiveness of our proposed designs in SAM-DETR and SAM-DETR++. Experiments for this ablation study are performed using ResNet-50 [76] as backbones under the 12-epoch (1x) training scheme.

Effect of Semantic-Aligned Matching. The first row in Table 4.4 shows the detection result of the origianl DETR baseline [1]. As shown in Table 4.4, the proposed Semantic Aligner, together with any query re-sampling strategy, consistently improves the performance over the baseline. We highlight that even with the naive max-pooling re-sampling, AP and $AP_{0.5}$ significantly improves by 4.7% and 10.7%, respectively. The results validate our claim that the proposed semantic-aligned matching mechanism effectively eases the matching difficulty between object queries and their corresponding target features, thus accelerating the training convergence of DETR.

Effect of Semantic-Aligned Matching with Features from Representative Keypoints. As shown in Table 4.4, different object query re-sampling strategies lead to large variance in detection performance. Max-pooling performs clearly better than average-pooling, which suggests that object detection relies more on

TABLE 4.4: Ablation study on the design choices for SAM-DETR and SAM-DETR++. Results are obtained on MS COCO val 2017 under the 12-epoch (1x) learning schedule.

Semantics Aligner	Query Re-Sampling Strategy				Feature Reweighting	Remove Dropout	Multi-Scale Feature Fusion	AP	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L
	AvgPool	MaxPool	Keypoint x1	Keypoints x8									
								22.3	39.5	22.2	6.6	22.8	36.6
✓	✓							25.2	48.9	23.3	8.9	26.4	41.3
✓		✓						27.0	50.2	25.8	10.3	28.0	43.9
✓			✓					28.6	50.3	28.1	12.4	31.2	44.4
✓			✓		✓			30.3	52.0	29.8	12.4	32.8	47.3
✓				✓				32.0	53.4	32.8	13.5	35.3	49.2
✓				✓	✓			33.1	54.2	33.7	13.9	36.5	51.7
✓				✓	✓	✓		34.2	55.8	35.3	15.0	37.7	52.5
						✓	✓	29.1	53.2	28.6	11.5	31.8	46.0
✓				✓	✓	✓	✓	**41.9**	**60.5**	**45.3**	**24.6**	**45.5**	**57.4**

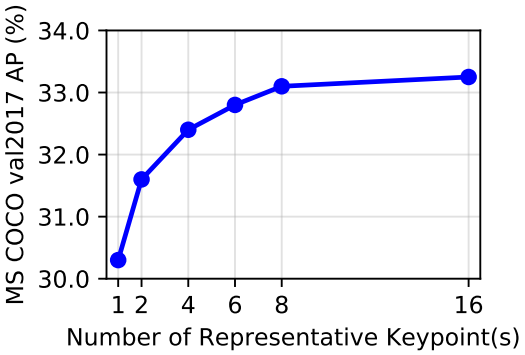


FIGURE 4.8: Ablation study on the number of searched representative keypoint(s). Results are obtained on MS COCO val 2017 with the 12-epoch (1x) training schedule without multi-scale feature fusion and removing dropout.

salient features rather than treating all features within reference boxes equally. This motivates us to explicitly search representative keypoints and employ their features for the introduced semantic-aligned matching mechanism. Results show that searching just one keypoint and re-sampling its features as new object queries outperforms the naive re-sampling strategies (AvgPool and MaxPool). Furthermore, searching multiple representative keypoints can naturally work with the multi-head attention mechanism [36] to further strengthen the representation capability of the re-sampled new object queries produced by Semantics Aligner and achieve superior performance. Fig. 4.8 also studies the effect of the number of searched representative keypoint(s) per object query. As Fig. 4.8 shows, the performance increases as the number of keypoints increases and saturates at 8. Therefore, we set the number of representative keypoints at 8 by default in SAM-DETR as well as SAM-DETR++.

TABLE 4.5: Ablation study on the representative keypoint search range. Results are obtained on MS COCO val 2017 with the 12-epoch (1x) training schedule without multi-scale feature fusion and removing dropout.

Representative Keypoint Search Range		AP	$AP_{0.5}$	$AP_{0.75}$
within ref box	within image			
✓		33.1	54.2	33.7
	✓	30.0	52.3	29.2

Searching within Boxes *vs.* Searching within Images. As introduced in Section 4.2.2, representative keypoints are searched within the corresponding reference boxes. We also evaluate the performance when allowing representative keypoints outside their reference boxes. As shown in Table 4.5, searching representative keypoints at the image scale degrades the performance. We suspect the performance drop is due to increased difficulty for matching with a larger search space. It is noteworthy that the original DETR's object queries do not have explicit search ranges, while our proposed SAM-DETR and SAM-DETR++ model learnable reference boxes with interpretable meanings, which effectively narrow down the search space, resulting in accelerated convergence.

Effect of Reweighting by Preceding Query Embeddings. As discussed in Section 4.2.2, previous object queries' embeddings contain helpful information for object detection, which cannot be directly leveraged due to the introduced re-sampling process. As a workaround, we propose to perform feature reweighting on re-sampled query embeddings based on previous query embeddings. This incorporates information from previous query embeddings while does not impede the aligned semantics. As shown in Table 4.4, the proposed feature reweighting mechanism consistently boosts performance, indicating its effectiveness.

Effect of Removing Dropout in Transformer. As shown in Table 4.4, the simple tweak of removing dropout [105] in Transformer [36] for object detection increases the performance of SAM-DETR without involving multi-scale feature fusion by 1.1% AP, at no extra computational cost.

Effect of Multi-Scale Feature Fusion with Aligned Semantics. As shown in Table 4.4, on top of SAM-DETR (with dropout removed), incorporating multi-scale feature fusion improves the detection performance by a considerable margin of 7.7% AP. This verifies that multi-scale feature fusion effectively reduces the

complexity of representing objects of different sizes and can adaptively choose appropriate feature scales for object representation, leading to further performance gain.

It is noteworthy that multi-scale feature fusion highly depends on our proposed semantic-aligned matching mechanism. As shown in Table 4.4, performing multi-scale feature fusion without our proposed semantic-aligned matching only yields a poor performance of 29.1% AP (-12.8% AP compared with SAM-DETR++). This is because there exists inevitable unaligned semantics across different feature scales. Without imposing aligned semantics, directly fusing multi-scale features that are projected into different feature embedding spaces (*i.e.*, with unaligned semantics) causes extra matching difficulty in cross-attention, as explained in Section 4.1 and Fig. 4.1.

4.3.6 Further Discussions

On the Compatibility with SMCA-DETR [84] and DN-DETR [85]. One of the key advantages of the proposed SAM-DETR and SAM-DETR++ is their excellent compatibility, which we demonstrate by integrating it with SMCA-DETR [84] and DN-DETR [85] and achieving superior detection performance. The reason behind their excellent compatibility is that each of them effectively accelerates the convergence of DETR from distinct perspectives, thus complementing each other. Concretely, SMCA-DETR [84] accelerates DETR's convergence by imposing strong spatial constraints for the cross-attention module, in which each object query is limited to attend to a specific region adaptively. SMCA-DETR [84] effectively reduces the search space for each object query in cross-attention, thus improving the training convergence. DN-DETR [85] proposes a de-noising training strategy to mitigate the instability of bipartite graph matching that causes inconsistent optimization goals in DETR's early training stages. With its proposed de-noising training strategy, the optimization objectives of DETR become consistent even in early training stages, which accelerates DETR's training convergence. Unlike the above two methods, our methods aim to reduce the matching difficulty between object query and encoded image features by enforcing aligned semantics, which encourages each object query to attend to those features with similar semantics. We demonstrate that with adequately addressed factors that obstruct convergence,

TABLE 4.6: Learnable reference boxes in SAM-DETR++ are not sentitive to gaps across different datasets.

Ref. Box	initialization	from scratch	pre-trained on MS COCO	random
	trainable?	✓	×	×
mAP@0.5 (%)		79.6	79.5	79.3

Transformer-based detectors do not fall behind conventional ConvNet-based detectors [21, 35, 37] in terms of convergence speed, with even superior performance and simpler pipelines.

Relevance and Difference with Sparse R-CNN [108]. SAM-DETR and SAM-DETR++ encode instance-level information with reference boxes and object queries, which have certain similarities to the proposal boxes and proposal features in Sparse R-CNN [108]. Besides, they all leverage RoIAlign [23] to pool region features. However, our methods are fundamentally distinct from Sparse R-CNN. As a member of the R-CNN family, Sparse R-CNN directly feeds the pooled features to a heavy R-CNN head to produce region-wise detection results. In contrast, our proposed SAM-DETR and SAM-DETR++ search and extract objects' salient features from the pooled region features using a lightweight network. The extracted features are fed to the Transformer modules for global predictions with accelerated convergence.

Are learnt reference boxes sensitive to gaps across datasets? The learned reference boxes encode statistical information for object distribution of specific datasets. To study whether these reference boxes affect generalization across datasets, we train and evaluate SAM-DETR++ w/ SMCA on Pascal VOC [5] (with notably different statistics from COCO [6]) over three setups for reference boxes: *(i)* learning from scratch on Pascal VOC, *(ii)* inheriting from COCO and remaining fixed, and *(iii)* remaining fixed from random initialization. Except for the reference boxes, all other parameters are trained on Pascal VOC. We use Pascal VOC trainval 07+12 for training and use test 07 for evaluation. Results in Table 4.6 show that the learned reference boxes generalize well across datasets. Even with totally random reference boxes, our method can still deliver satisfactory detection accuracy. This is because *(i)* the reference boxes are dense enough to cover most image regions, and *(ii)* SAM-DETR and SAM-DETR++ involve multiple stages of box adjustment, thus initial reference boxes do not have a clear impact on final predictions.

4.4 Conclusion

This chapter presents SAM-DETR and SAM-DETR++ to accelerate the convergence of DETR. The core of SAM-DETR is a plug-and-play module that semantically aligns object queries and encoded image features to facilitate the matching procedure between them. It also explicitly searches multiple representative keypoints with the most discriminative features for semantic-aligned matching. Besides, on the basis of SAM-DETR, SAM-DETR++ can further fuse multi-scale features in a coarse-to-fine manner via the semantic-aligned matching mechanism. The fusion of multi-scale features effectively helps represent objects of different scales in a 'divide-and-conquer' manner, further lowering the representation complexity and improving convergence. By simply introducing a plug-and-play module, our proposed SAM-DETR and SAM-DETR++ accelerate DETR's convergence from a unique perspective, and thus can be easily integrated with existing convergence solutions to boost performance to a greater extent. On the MS COCO benchmark, the fully-fledged SAM-DETR++ achieves 44.8% AP with only 12 training epochs, outperforming Faster R-CNN by a large margin. It also achieves state-of-the-art detection accuracy among Transformer-based detectors. We demonstrate that Transformer-based detectors do not fall behind conventional ConvNet-based detectors in terms of convergence speed, with even superior performance and simpler pipelines. We hope this work paves the way for more comprehensive research and applications of Transformer-based object detectors.

Chapter 5

Meta-DETR: Image-Level Few-Shot Object Detection with Exploitation of Inter-Class Correlation [1]

5.1 Introduction

Computer vision has experienced significant progress in recent years with the advancement of deep neural networks. However, there still exists a huge gap between current computer vision techniques and the human visual system in learning new concepts from very few examples: most existing methods require a large amount of annotated samples, while humans can effortlessly recognize a new concept even with just a glimpse of it [111]. Such human-like capability to generalize from limited examples is still absent in most modern machine vision systems.

Modern object detectors [1, 17, 20, 21, 26, 28, 29, 31, 34, 35, 37–39, 41, 42, 70, 84, 104, 108] are primarily developed upon deep neural networks. Therefore, most of these modern object detectors [1, 17, 20, 21, 26, 28, 29, 31, 34, 35, 37–39, 41, 42,

[1] The work in this chapter has been published as Gongjie Zhang, Zhipeng Luo, Kaiwen Cui, Shijian Lu, and Eric P. Xing. "Meta-DETR: Image-Level Few-Shot Detection with Inter-Class Correlation Exploitation." IEEE Transactions on Pattern Analysis and Machine Intelligence (**T-PAMI**), 2022. (DOI: 10.1109/TPAMI.2022.3195735) [43]

70, 84, 104, 108] are heavily dependent on human supervision in the form of large amounts of annotated training samples. However, such large-scale and annotated datasets are not always available due to expensive human labeling costs and/or difficulty in data acquisition [112–114]. Under such circumstances where only a few annotated samples are available, modern object detectors usually suffer from a huge performance drop and fail to deliver satisfactory detection accuracy.

This chapter explores the challenging task of few-shot object detection to adapt to such challenging scenarios when sufficient training samples are not available. Few-shot object detection targets detecting novel objects with only a few training samples. With minimal supervision on novel classes, the key to few-shot object detection is to learn transferable knowledge from base classes and generalize it to novel classes. To this end, many studies [27, 62, 64, 115, 116] incorporate meta-learning into generic region-based object detection frameworks, mostly Faster R-CNN [21], and have achieved promising results.

Despite their success, there still exist two underlying limitations that hinder better exploitation of base-class knowledge, as illustrated in Fig. 5.2. *First*, region-based detection frameworks rely on region proposals to produce final predictions, thus are sensitive to low-quality region proposals. However, as investigated by [115] and [117], it is not easy to produce high-quality region proposals for novel classes with limited supervision under the few-shot detection setups. Such a gap in the quality of region proposals obstructs the generalization from base classes to novel classes. *Second*, most existing meta-learning-based approaches [27, 62, 64, 115] adopt 'feature reweighting' or its variants to aggregate query and support features, which can only deal with one support class (*i.e.*, target class to detect) at a time and essentially treat each support class independently. Without seeing multiple classes within a single feedforward, they largely overlook the important inter-class correlation among different support classes. This limits the ability to distinguish similar classes (*e.g.*, distinguishing from cows and horses) and to generalize from related classes (*e.g.*, learning to detect cows by generalizing from detecting sheep).

To mitigate the above limitations, we design Meta-DETR, an innovative few-shot object detector that performs pure image-level prediction and at the same time exploits the inter-class correlation among different classes. Fig. 5.1 illustrates its major differences with prior designs. To our best knowledge, this is the first work that identifies the constraint caused by region-based detection under the few-shot

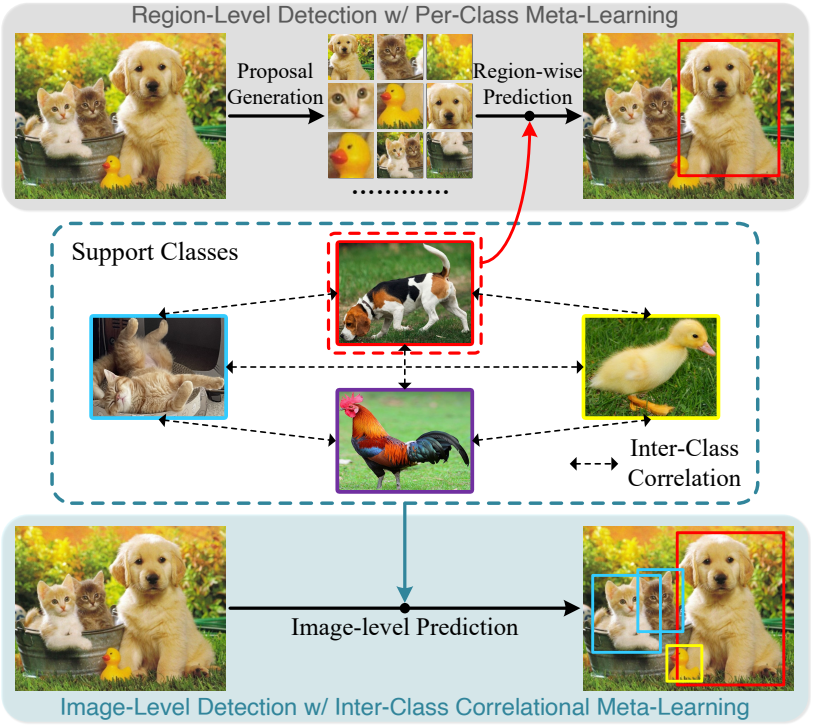
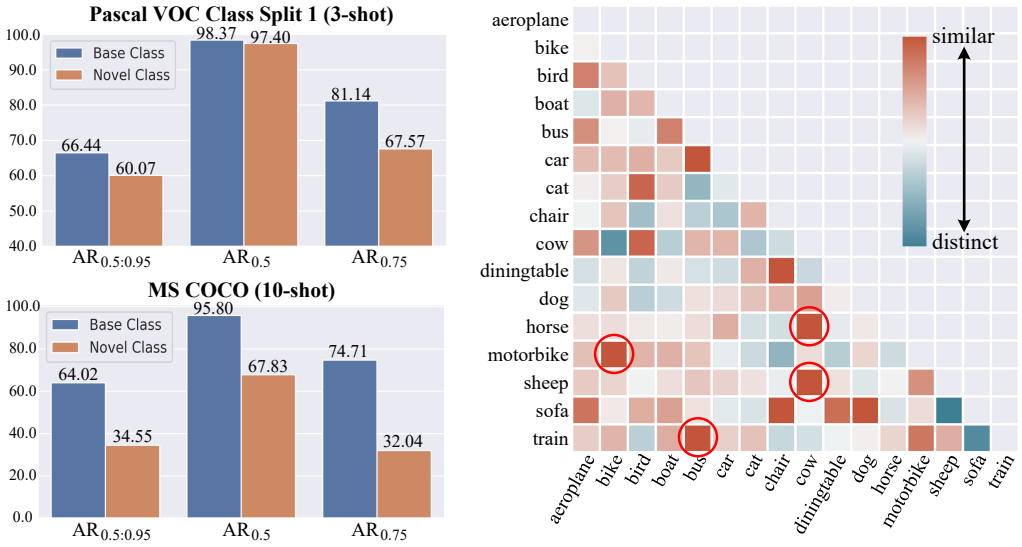


FIGURE 5.1: Comparison of few-shot object detection pipelines. Prior studies (upper part) perform region-level detection, which are often constrained by inaccurate region proposals for novel classes. Besides, they can only deal with one support class at one go and overlook the correlation among different classes. The proposed Meta-DETR (lower part) works at image level without any proposals. It captures inter-class correlation by learning from multiple support classes simultaneously, which suppresses confusion among similar classes and enhances model generalization greatly.

setups and explores to address few-shot object detection with DETR-based detection frameworks, which can skip proposal generation and directly perform detection at image level. With image-level prediction, Meta-DETR fully bypasses the constraint of inaccurate region proposals as in prevalent few-shot detection frameworks. In addition, the introduced inter-class correlational meta-learning strategy enables Meta-DETR to attend to multiple support classes at one go instead of class-by-class meta-learning with repeated runs as in most existing methods. By integrating detection tasks that involve multiple classes into meta-learning, Meta-DETR can explicitly leverage the inter-class correlation, including the inter-class commonality to facilitate generalization among related classes and the inter-class uniqueness to reduce mis-classification among similar classes.

In summary, the contributions of Meta-DETR presented in this chapter are threefold. *First*, we identify the quality gap of proposals for base and novel classes



(a) Quality Gap in Region Proposals

(b) Cosine Similarity of Class Prototypes

(c) Mis-classification of Related Classes

FIGURE 5.2: Existing few-shot detection frameworks tend to suffer from inaccurate region proposals and negligence of inter-class correlation. Due to very limited training samples for novel classes, the proposal quality (measured by Average Recall on top 1000 proposals) for novel classes is clearly lower than that of base classes as illustrated in (a). This hinders the knowledge generalization to novel classes. Additionally, object classes with similar appearances are highly correlated in feature space such as 'cow *vs.* horse' and 'motorbike *vs.* bike' as illustrated in (b), which tend to be mis-classified if the learning does not incorporate the correlation among them as illustrated in (c).

in region-based prediction, and propose Meta-DETR to address few-shot object detection. Being the first pure image-level few-shot detector, Meta-DETR fully circumvents the gap of inaccurate proposals for novel-class objects, thus enabling

better generalization to novel classes. *Second*, we design a novel correlational meta-learning strategy, which can deal with multiple support classes simultaneously. It effectively exploits inter-class correlation among different classes, thus greatly reducing mis-classification and enhancing model generalization. *Third*, extensive experiments show that, without bells and whistles, the proposed Meta-DETR consistently outperforms state-of-the-art methods by large margins on the task of detecting novel objects with just a few samples.

5.2 Preliminaries

5.2.1 Few-Shot Object Detection

Given two sets of classes $\mathcal{C}_{\text{base}}$ and $\mathcal{C}_{\text{novel}}$, where $\mathcal{C}_{\text{base}} \cap \mathcal{C}_{\text{novel}} = \varnothing$, a few-shot object detector aims at detecting objects of $\mathcal{C}_{\text{base}} \cup \mathcal{C}_{\text{novel}}$ by learning from a base dataset $\mathcal{D}_{\text{base}}$ with abundant annotated objects of $\mathcal{C}_{\text{base}}$ and a novel dataset $\mathcal{D}_{\text{novel}}$ with very few annotated objects of $\mathcal{C}_{\text{novel}}$. In the task of K-shot object detection, there are exactly K annotated objects for each novel class in $\mathcal{D}_{\text{novel}}$.

Existing works on few-shot object detection can be mainly categorized into two paradigms: transfer learning and meta-learning. Methods with transfer learning mainly include LSTD [61], TFA [63], MPSR [118], and FSCE [119], where novel concepts are learned via fine-tuning. Differently, methods with meta-learning [27, 62, 64, 115, 116, 120, 121] extract knowledge that can generalize across various tasks via 'learning to learn', *i.e.*, learning a class-agnostic predictor on various auxiliary tasks, in which target classes are dynamically conditioned on support images.

Our proposed Meta-DETR, which is to be detailed later, falls under the umbrella of meta-learning, but differs from existing approaches by achieving image-level meta-learning-based detection prediction and effectively leveraging the correlation among various support classes. To the best of our knowledge, Meta-DETR is the first work that incorporates meta-learning into the recently proposed DETR frameworks. It is also the pioneering work to explicitly integrate the inter-class correlation among support classes into few-shot object detection frameworks using meta-learning.

5.2.2 Rethinking Region-Based Detection Frameworks

Most existing few-shot object detectors are developed on top of Faster R-CNN [21], a region-based object detector, thanks to its robust performance and ease for optimization. Faster R-CNN first uses a Region Proposal Network (RPN) to generate region proposals and then performs region-wise predictions to produce final object detection predictions. However, by relying on region proposals to produce detection results, these existing few-shot object detectors are inevitably constrained by the inaccurate proposals for novel classes due to very limited supervision under the few-shot detection setups. As illustrated in Fig. 5.2(a), there is a clear gap in the quality of region proposals for base and novel classes, hindering region-based detection frameworks from exploiting base-class knowledge to generalize to novel classes. Though several studies [115, 117] attempt to acquire more accurate region proposals, this issue still remains as it is rooted in the region-based detection frameworks under the few-shot learning setups.

5.2.3 Rethinking Meta-Learning via Feature Reweighting

To meta-learn a class-agnostic detector that can generalize across various classes, most existing methods [27, 62, 64, 115] adopt 'feature reweighting' or its variants to aggregate query features with support class information, acquiring class-specific meta-features to detect objects corresponding to the support class. However, such meta-learning strategies can deal with only one support class within each feed-forward process, *i.e.*, C repeated runs are required to detect C support classes within each query image. More importantly, by treating each support class independently, 'feature reweighting' overlooks the essential inter-class correlation among different support classes. As shown in Fig. 5.2(b), many object classes with similar appearances are highly correlated. Intuitively, their correlation can effectively facilitate the distinction and the generalization among similar classes. However, as shown in Fig. 5.2(c), we observe that objects mis-classified as highly correlated classes constitute a major source of error due to the negligence of inter-class correlation in existing methods.

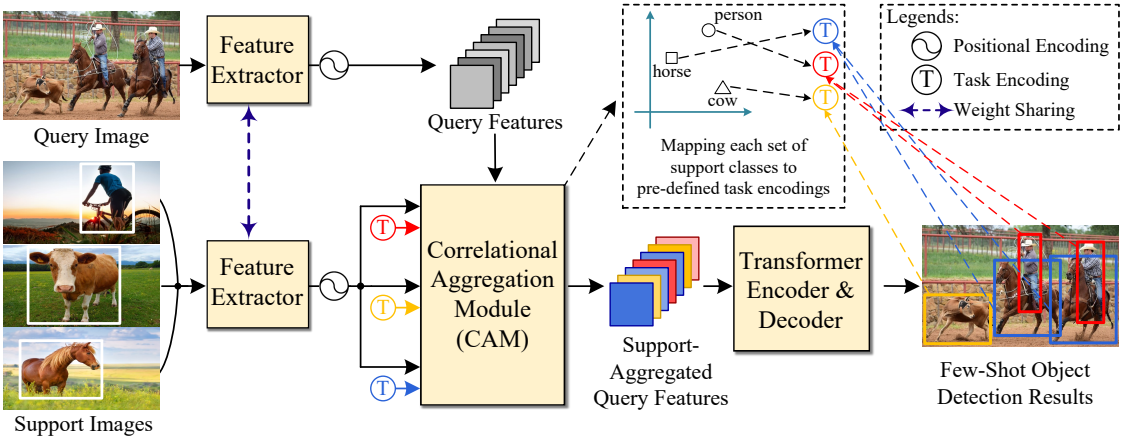


FIGURE 5.3: The overall framework of Meta-DETR. Query image and support images are processed by a weight-shared feature extractor to produce query features and support features. To leverage the inter-class correlation in meta-learning, a Correlational Aggregation Module (CAM) is designed, which first matches the query features with multiple support classes simultaneously and then introduces multiple task encodings (*i.e.*, the three illustrative Ⓣ of different colors) to differentiate these support classes. Finally, few-shot detection is achieved with a class-agnostic Transformer encoder and decoder that learns to predict objects' locations and their corresponding task encodings (instead of directly predicting objects' class labels).

5.3 Methodology

This section provides a detailed description of the proposed Meta-DETR, including its network architecture, training objective, as well as the learning and inference procedure.

5.3.1 Overview

Fig. 5.3 shows the architecture of the proposed Meta-DETR. Motivated by previous discussions, Meta-DETR employs the recently proposed Deformable DETR [38], a fully end-to-end Transformer-based [36] detector, as the basic detection framework. As Meta-DETR does not rely on predicted region proposals to make final predictions, it can fully bypass the constraint of inaccurate proposals on novel-class objects. Besides, thanks to the introduced correlational meta-learning, Meta-DETR can aggregate query features with multiple support classes simultaneously, thus capturing and leveraging the inter-class correlation among different classes to reduce mis-classification and boost generalization.

Given a query image and a set of support images with instance annotations, a weight-shared feature extractor first encodes them into the same feature space. Subsequently, a *Correlational Aggregation Module (CAM)*, which will be introduced later, performs simultaneous aggregation between the query features and the set of support classes. To differentiate between different support classes in a class-agnostic manner, CAM introduces a set of task encodings assigned to each support class. Finally, a transformer architecture detects objects by predicting their locations and corresponding task encodings. As the detection targets are dynamically determined by support classes and their mappings to task encodings, Meta-DETR is trained as a meta-learner to extract generalizable knowledge not specific to certain classes.

5.3.2 Inter-Class Correlational Meta-Learning

The *Correlational Aggregation Module (CAM)* is the key component in Meta-DETR to perform inter-class correlational meta-learning, which aggregates query features with support classes for the subsequent class-agnostic prediction. CAM differs from existing aggregation methods in that it can aggregate multiple support classes simultaneously, which enables it to capture their inter-class correlation to reduce mis-classification and enhance model generalization. Specifically, as illustrated in Fig. 5.4, the query and support features are first processed by a weight-shared multi-head attention module, encoding them into the same embedding space. Then the prototype for each support class is obtained by applying RoIAlign [23], followed by average pooling on the support features, where RoIAlign ensures that class prototypes are obtained from the relevant regions that contain corresponding support object instances. After that, CAM performs feature matching and encoding matching, which will be elaborated in the remainder of this subsection to match the query features with support class prototypes and task encodings, respectively. The matching results are summed together and fed to a feed-forward network (FFN) to produce the final output. Note that the support class prototypes are obtained in CAM before feature matching and encoding matching.

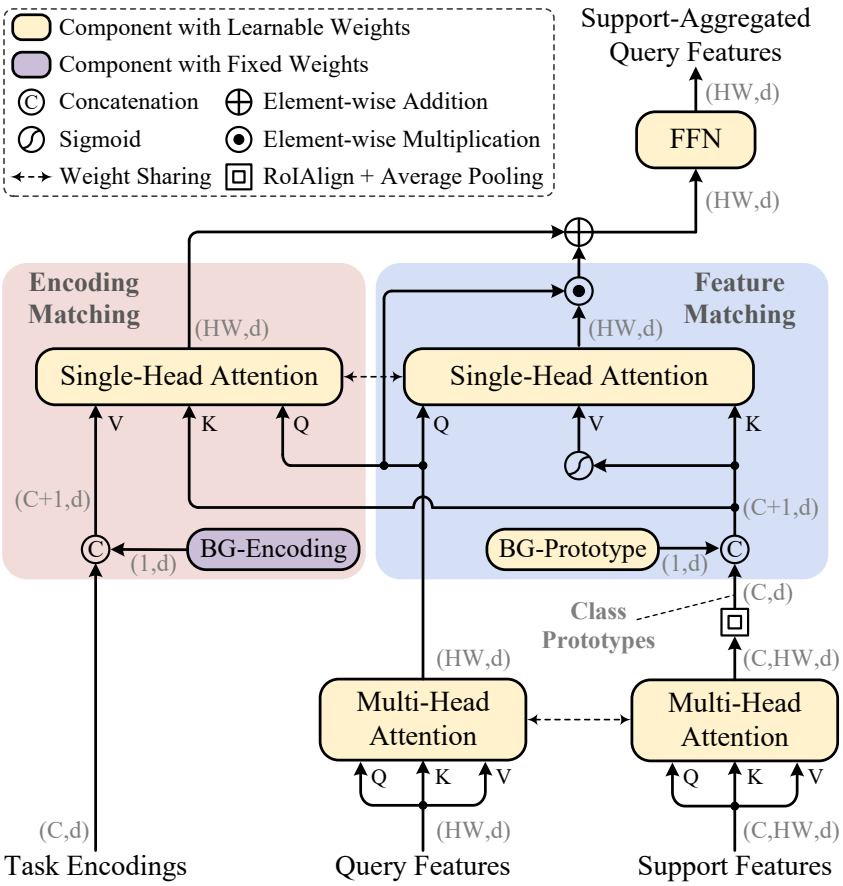


FIGURE 5.4: The architecture of the Correlational Aggregation Module (CAM). CAM first obtains class prototypes from support features. Then, it performs two matching processes: *Feature Matching* filters out query features that are unrelated to support classes, while *Encoding Matching* matches query features to a set of pre-defined task encodings that differentiate their corresponding support classes in a class-agnostic manner.

5.3.2.1 Feature Matching

Feature matching, which aims to filter out features irrelevant to support classes, is achieved by an attention mechanism with minor modifications. Specifically, given a query feature map $\mathbf{Q} \in \mathbb{R}^{HW \times d}$ and the support class prototypes $\mathbf{S} \in \mathbb{R}^{C \times d}$, where HW is the spatial size, C is the number of support classes, and d is the feature dimensionality, the matching coefficients are obtained via:

$$\mathbf{A} = \text{Attn}(\mathbf{Q}, \mathbf{S}) = \text{Softmax}(\frac{(\mathbf{QW})(\mathbf{SW})^{\text{T}}}{\sqrt{d}}), \tag{5.1}$$

where $\mathbf{W}$ is a linear projection shared by $\mathbf{Q}$ and $\mathbf{S}$, which ensures they are embedded into the same feature space. Subsequently, the output of the feature matching module can be obtained via:

$$\mathbf{Q_F} = \mathbf{A}\sigma(\mathbf{S}) \odot \mathbf{Q}, \tag{5.2}$$

where $\sigma(\cdot)$ denotes sigmoid function and $\odot$ denotes Hadamard product. $\sigma(\mathbf{S})$ serves as feature filters for each individual support class with the function of extracting only class-related features from query features. By applying the matching coefficients $\mathbf{A}$ to $\sigma(\mathbf{S})$, we have filters that can filter out query features that are not matched to any support class, producing a filtered query feature map $\mathbf{Q_F}$ that only highlights objects belonging to the given support classes.

5.3.2.2 Encoding Matching

To achieve correlational meta-learning, we introduce a set of pre-defined task encodings assigned to each support class and match query features to their corresponding task encodings, so that final predictions can be made on the task encodings instead of specific classes. We implement task encodings $\mathbf{T} \in \mathbb{R}^{C \times d}$ with sinusoidal functions, following the positional encodings of the Transformer [36]. Encoding matching uses the same matching coefficients as feature matching, and the matched encodings $\mathbf{Q_E}$ are obtained via:

$$\mathbf{Q_E} = \mathbf{AT}. \tag{5.3}$$

5.3.2.3 Modeling Background for Open-Set Prediction

Object detection features an open-set setup where background, which does not belong to any of the target classes, often takes up most of the spatial locations in a query image. Therefore, as shown in Fig. 5.4, we additionally introduce a learnable prototype and a corresponding task encoding (fixed to zeros), denoted as BG-Prototype and BG-Encoding respectively, to explicitly model the background class. This eliminates the matching ambiguity when query does not match any of the given support classes.

5.3.3 Network Optimization

5.3.3.1 Target Generation for Meta-Learning

We let N denote the fixed number of object queries, which means Meta-DETR infers N predictions within a single feed-forward process. Let x_{query} denote the query image, and $y = \{y_i\}_{i=1}^{N}$ denote the ground truth objects within the query image, where y is a set of size N. When y_i indicates an object, $y_i = (c_i, b_i)$, where c_i denotes the target class label and b_i denotes the bounding box of the object. When y_i indicates no object, $y_i = (\varnothing, \varnothing)$.

Meta-DETR dynamically conditions its detection targets on the sampled support classes and their mappings to the task encodings. As discussed in Section 5.3.1, Meta-DETR predicts over C support classes (*i.e.*, target classes) simultaneously. The C support classes are randomly sampled, denoted as $c_{\text{supp}} = \{s_i\}_{i=1}^{C}$. Besides, these support classes are further mapped to a set of task encodings. We denote the mapping function from the labels of support classes to the labels of task encodings as $\chi(\cdot)$. A specific case of $\chi(\cdot)$ can be formulated as:

$$\chi(s_i) = i \qquad i \in \{1, 2, \cdots, C\}. \tag{5.4}$$

Note that the exact format of the mapping function $\chi(\cdot)$ does not matter. Then, the detection targets of Meta-DETR can be formulated as:

$$y' = \{y'_i\}_{i=1}^{N} = \{(c'_i, b'_i)\}_{i=1}^{N} = \{\psi(y_i, c_{\text{supp}})\}_{i=1}^{N}, \tag{5.5}$$

where $\psi(y_i, c_{\text{supp}})$ acts to remove annotations of irrelevant objects (objects with labels not in c_{supp}) and to map the labels of target classes to the labels of the corresponding task encodings, which can be formulated as:

$$\psi(y_i, c_{\text{supp}}) = \begin{cases} (\varnothing, \varnothing), & \text{if } y_i = (\varnothing, \varnothing) \text{ or } \mathrm{c_i} \notin \mathrm{c_{supp}} \\ (\chi(c_i), b_i), & \text{if } c_i \in c_{\text{supp}}. \end{cases} \tag{5.6}$$

Note that y' can completely consist of $(\varnothing, \varnothing)$ when there is no objects that belong to the provided support classes.

5.3.3.2 Loss Function

Assume the N predictions for target class made by Meta-DETR are $\hat{y}=\{\hat{y}_i\}_{i=1}^{N}=\{(\hat{c}_i,\hat{b}_i)\}_{i=1}^{N}$. We adopt a pair-wise matching loss $\mathcal{L}_{\text{match}}(y'_i,\hat{y}_{\sigma(i)})$ to search for a bipartite matching between $\hat{y}$ and y' with the lowest cost:

$$\hat{\sigma}=\arg\min_{\sigma}\sum\nolimits_{i=1}^{N}\mathcal{L}_{\text{match}}(y'_i,\hat{y}_{\sigma(i)}), \tag{5.7}$$

where σ denotes a permutation of N elements, and $\hat{\sigma}$ denotes the optimal assignment between predictions and targets. Since the matching should consider both classification and localization, the matching loss is defined as:

$$\begin{aligned}\mathcal{L}_{\text{match}}(y'_i,\hat{y}_{\sigma(i)})=&\mathbb{1}_{\{c'_i\neq\varnothing\}}\mathcal{L}_{\text{cls}}(c'_i,\hat{c}_{\sigma(i)})+\\&\mathbb{1}_{\{c'_i\neq\varnothing\}}\mathcal{L}_{\text{box}}(b'_i,\hat{b}_{\sigma(i)})\quad.\end{aligned} \tag{5.8}$$

With the optimal assignment $\hat{\sigma}$ obtained with Eq. 5.7 and Eq. 5.8, we optimize the network using the following loss function:

$$\mathcal{L}(y',\hat{y})=\sum_{i=1}^{N}\left[\mathcal{L}_{\text{cls}}(c'_i,\hat{c}_{\hat{\sigma}(i)})+\mathbb{1}_{\{c'_i\neq\varnothing\}}\mathcal{L}_{\text{box}}(b'_i,\hat{b}_{\hat{\sigma}(i)})\right], \tag{5.9}$$

where we adopt sigmoid focal loss [32] for $\mathcal{L}_{\text{cls}}$ and adopt a linear combination of $\ell 1$ loss and GIoU loss [122] for $\mathcal{L}_{\text{box}}$. Similar to DETR [1] and Deformable DETR [38], $\mathcal{L}(y',\hat{y})$ is applied to every layer of the transformer decoder.

Following Meta R-CNN [27], we introduce a cosine similarity cross-entropy loss [123] to classify the class prototypes obtained by our designed CAM. It encourages prototypes of different classes to be distinguished from each other.

5.3.3.3 Two-Stage Training Procedure

The training procedure consists of two stages. The first stage is *base training stage*. During this stage, the model is trained on the base dataset $\mathcal{D}_{\text{base}}$ with abundant training samples for each base class. The second stage is *few-shot fine-tuning stage*. In this stage, we train the model on both base and novel classes with limited training samples. Only K object instances are available for each novel category

in K-shot object detection. Following prior works [27, 63, 64], we also include objects from base classes to prevent performance drop for base classes. In both *base training* and *few-shot fine-tuning* stages, the whole network is optimized in an end-to-end manner with the same training objective.

5.3.4 Efficient Inference Procedure

Unlike the training stage, there is no need to repeatedly sample support images and extract their features with the feature extractor. We can first compute the prototype for each support class once and for all, then directly use them for every query image to predict. This promises efficient inference of our proposed Meta-DETR.

5.4 Experiments

5.4.1 Datasets and Evaluation Metrics

We follow the well-established data setups for few-shot object detection [27, 62–64, 120]. Concretely, two widely used few-shot object detection benchmarks are adopted in our experiments.

- **Pascal VOC** [5] is a commonly used dataset for object detection that consists of images with object annotations of 20 classes. We use *trainval 07+12* for training and perform evaluations on *test 07*. We use 3 novel / base class splits, *i.e.*, ("bird", "bus", "cow", "motorbike", "sofa" / others), ("aeroplane", "bottle", "cow", "horse", "sofa" / others), and ("boat", "cat", "motorbike", "sheep", "sofa" / others). The number of shots is set to 1, 2, 3, 5 and 10. Mean average precision at IoU threshold 0.5 (mAP@0.5) is used as the evaluation metric. Results are averaged over 10 randomly sampled support datasets.

- **MS COCO** [6] is a more challenging object detection dataset, which contains 80 classes including those 20 classes in Pascal VOC. We adopt the 20 shared classes as novel classes, and adopt the remaining 60 classes as base classes.

The number of shots is set to 1, 3, 5, 10, and 30. We use *train 2017* for training, and perform evaluations on *val 2017*. Standard evaluation metrics for MS COCO are adopted. Results are averaged over 5 randomly sampled support datasets.

5.4.2 Implementation Details

We adopt the commonly used ResNet-101 [76] as the feature extractor. The network architectures and hyper-parameters remain the same as Deformable DETR [38]. We implement our model in single-scale version for fair comparison with other works. We also follow FsDetView [64] to implement the aggregation with a slightly more complex scheme compared with solely feature reweighting. We train our model with 8 x Nvidia V100 GPUs, using the AdamW [106, 107] optimizer with an initial learning rate of 2×10^{-4} and a weight decay of 1×10^{-4}. Batch size is set to 32. In the base training stage, we train the model for 50 and 25 epochs for Pascal VOC and MS COCO, respectively. Learning rate is decayed at the 45^{th} and 20^{th} epoch by 0.1. In the few-shot fine-tuning stage, the same settings are applied to fine-tune the model until convergence.

5.4.3 Comparison with State-of-the-Art Methods

5.4.3.1 Pascal VOC

Table 5.1 shows the few-shot detection performance for novel classes of Pascal VOC. It can be seen that Meta-DETR consistently outperforms existing methods across various setups. With multiple runs over randomly sampled support datasets to reduce randomness, Meta-DETR achieves the best average performance across all setups, with a large margin of $+4.6\%$ overall mAP compared with the second-best. The strong performance demonstrates the superiority and robustness of our proposed Meta-DETR.

We also present results taking base classes into consideration in Table 5.2. While achieving good performance for novel classes with limited training samples, Meta-DETR can still detect objects of base classes with competitive performance. TFA [63]

TABLE 5.1: Few-shot detection performance (mAP@0.5) on Pascal VOC for novel classes. "‡" indicates methods using multi-scale features. "△" indicates re-evaluated results using official codes. "⊎" indicates usage of external data.

	Class Split 1					Class Split 2					Class Split 3					*Avg.*
Method \ Shots	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10	
• *Results over a single run:*																
LSTD [61]	8.2	1.0	12.4	29.1	38.5	11.4	3.8	5.0	15.7	31.0	12.6	8.5	15.0	27.3	36.3	17.1
RepMet [124] ‡	26.1	32.9	34.4	38.6	41.3	17.2	22.1	23.4	28.3	35.8	27.5	31.1	31.5	34.4	37.2	30.8
Meta-YOLO [62]	14.8	15.5	26.7	33.9	47.2	15.7	15.3	22.7	30.1	40.5	21.3	25.6	28.4	42.8	45.9	28.4
Meta Det [120]	18.9	20.6	30.2	36.8	49.6	21.8	23.1	27.8	31.7	43.0	20.6	23.9	29.4	43.9	44.1	31.0
Meta R-CNN [27]	19.9	25.5	35.0	45.7	51.5	10.4	19.4	29.6	34.8	45.4	14.3	18.2	27.5	41.2	48.1	31.1
TFA w/ fc [63] ‡	36.8	29.1	43.6	55.7	57.0	18.2	29.0	33.4	35.5	39.0	27.7	33.6	42.5	48.7	50.2	38.7
TFA w/ cos [63] ‡	39.8	36.1	44.7	55.7	56.0	23.5	26.9	34.1	35.1	39.1	30.8	34.8	42.8	49.5	49.8	39.9
MPSR [118] ‡	41.7	43.1	51.4	55.2	61.8	24.4	29.5	39.2	39.9	47.8	35.6	40.6	42.3	48.0	49.7	43.3
TFA w/ cos + Halluc [66] ‡	45.1	44.0	44.7	55.0	55.9	23.2	27.5	35.1	34.9	39.0	30.5	35.1	41.4	49.0	49.3	40.6
Retentive R-CNN [125] ‡	42.4	45.8	45.9	53.7	56.1	21.7	27.8	35.2	37.0	40.3	30.2	37.6	43.0	49.7	50.1	41.1
CME [126] ‡	41.5	47.5	50.4	58.2	60.9	27.2	30.2	41.4	42.5	46.8	34.3	39.6	45.1	48.3	51.5	44.4
SRR-FSD [65] ‡⊎	**47.8**	50.5	51.3	55.2	56.8	32.5	35.3	39.1	40.8	43.8	40.1	41.5	44.3	46.9	46.4	44.8
FSCE [119] ‡	44.2	43.8	51.4	**61.9**	63.4	27.3	29.5	43.5	44.2	50.2	37.2	41.9	47.5	54.6	58.5	46.6
Meta-DETR (Ours)	40.6	**51.4**	**58.0**	59.2	**63.6**	**37.0**	**36.6**	**43.7**	**49.1**	**54.6**	**41.6**	**45.9**	**52.7**	**58.9**	**60.6**	**50.2**
• *Results averaged over multiple random runs:*																
FRCN+ft-full [21] ‡	9.9	15.6	21.6	28.0	35.6	9.4	13.8	17.4	21.9	29.8	8.1	13.9	19.0	23.9	31.0	19.9
Deformable-DETR+ft-full [38] ‡	5.6	13.3	21.7	34.2	45.0	10.9	13.0	18.4	27.3	39.4	7.3	16.6	20.8	32.2	41.8	23.2
TFA w/ fc [63] ‡	22.9	34.5	40.4	46.7	52.0	16.9	26.4	30.5	34.6	39.7	15.7	27.2	34.7	40.8	44.6	33.8
TFA w/ cos [63] ‡	25.3	36.4	42.1	47.9	52.8	18.3	27.5	30.9	34.1	39.5	17.9	27.2	34.3	40.8	45.6	34.7
FsDetView [64]	24.2	35.3	42.2	49.1	57.4	21.6	24.6	31.9	37.0	45.7	21.2	30.0	37.2	43.8	49.6	36.7
MPSR [118] ‡△	34.7	42.6	46.1	49.4	56.7	22.6	30.5	31.0	36.7	43.3	27.5	32.5	38.2	44.6	50.0	39.1
DCNet [116] ‡	33.9	37.4	43.7	51.1	59.6	23.2	24.8	30.6	36.7	46.6	32.3	34.9	39.7	42.6	50.7	39.2
FSCE [119] ‡	32.9	44.0	46.8	52.9	59.7	23.7	30.6	**38.4**	43.0	48.5	22.6	33.4	39.5	47.3	54.0	41.2
Meta-DETR (Ours)	**35.1**	**49.0**	**53.2**	**57.4**	**62.0**	**27.9**	**32.3**	**38.4**	**43.2**	**51.8**	**34.9**	**41.8**	**47.1**	**54.1**	**58.2**	**45.8**

TABLE 5.2: Few-shot detection performance (mAP@0.5) on Pascal VOC class split 1 for both base and novel classes. "§" indicates results averaged over multiple random runs.

	Base Classes				Novel Classes			
Method \ Shots	1	3	5	10	1	3	5	10
Meta-YOLO [62]	66.4	64.8	63.4	63.6	14.8	26.7	33.9	47.2
FsDetView [64] §	64.2	69.4	69.8	71.1	24.2	42.2	49.1	57.4
TFA w/ cos [63] §	**77.6**	**77.3**	**77.4**	**77.5**	25.3	42.1	47.9	52.9
MPSR [118] §	60.6	65.9	68.2	69.8	34.7	46.1	49.4	56.7
FSCE [119] §	75.5	73.7	75.0	75.2	32.9	46.8	52.9	59.7
Meta-DETR (Ours) §	67.2	70.0	73.0	73.5	**35.1**	**53.2**	**57.4**	**62.0**

produces outstanding performance for base classes since it only fine-tunes detector's last layer, thus having relatively constrained capacity in generalizing on novel classes. We would highlight that our proposed Meta-DETR achieves the best base-class and novel-class performance among all compared methods using meta-learning (*i.e.*, Meta-YOLO [62] and FsDetView [64]).

TABLE 5.3: Few-shot detection performance on MS COCO for novel classes. "‡" indicates methods using multi-scale features. "§" indicates results averaged over multiple runs. "⊎" indicates usage of external data.

Shot	Method	$AP_{0.5:0.95}$	$AP_{0.5}$	$AP_{0.75}$
1	FRCN+ft-full [21] ‡ §	1.7	3.3	1.6
	Deformable-DETR+ft-full [38] §	1.8	3.1	1.8
	TFA w/ cos [63] ‡ §	1.9	3.8	1.7
	TFA w/ cos + Halluc [66] ‡	3.8	6.5	4.3
	Meta-DETR (Ours) §	**7.5**	**12.5**	**7.7**
3	FRCN+ft-full [21] ‡ §	3.7	7.1	3.5
	Deformable-DETR+ft-full [38] §	4.9	7.8	5.1
	TFA w/ cos [63] ‡ §	5.1	9.9	4.8
	TFA w/ cos + Halluc [66] ‡	6.9	12.6	7.0
	Meta-DETR (Ours) §	**13.5**	**21.7**	**14.0**
5	FRCN+ft-full [21] ‡ §	4.6	8.7	4.4
	Deformable-DETR+ft-full [38] §	7.4	12.3	7.7
	TFA w/ cos [63] ‡ §	7.0	13.3	6.5
	FsDetView [64] §	10.7	24.5	6.7
	Meta-DETR (Ours) §	**15.4**	**25.0**	**15.8**
10	FRCN+ft-full [21] ‡ §	5.5	10.0	5.5
	Deformable-DETR+ft-full [38] §	11.7	19.6	12.1
	Meta-YOLO [62]	5.6	12.3	4.6
	Meta Det [120]	7.1	14.6	6.1
	Meta R-CNN [27]	8.7	19.1	6.6
	TFA w/ cos [63] ‡ §	9.1	17.1	8.8
	FSOD [115]	12.0	22.4	11.8
	FsDetView [64] §	12.5	27.3	9.8
	MPSR [118] ‡	9.8	17.9	9.7
	SRR-FSD [65] ‡ ⊎	11.3	23.0	9.8
	CME [126] ‡	15.1	24.6	16.4
	DCNet [116] ‡ §	12.8	23.4	11.2
	FSCE [119] ‡ §	11.1	-	9.8
	Meta-DETR (Ours) §	**19.0**	**30.5**	**19.7**
30	FRCN+ft-full [21] ‡ §	7.4	13.1	7.4
	Deformable-DETR+ft-full [38] §	16.3	27.2	16.7
	Meta-YOLO [62]	9.1	19.0	7.6
	Meta Det [120]	11.3	21.7	8.1
	Meta R-CNN [27]	12.4	25.3	10.8
	TFA w/ cos [63] ‡ §	12.1	22.0	12.0
	FsDetView [64] §	14.7	30.6	12.2
	MPSR [118] ‡	14.1	25.4	14.2
	SRR-FSD [65] ‡ ⊎	14.7	29.2	13.5
	CME [126] ‡	16.9	28.0	17.8
	DCNet [116] ‡ §	18.6	32.6	17.5
	FSCE [119] ‡ §	15.3	-	14.2
	Meta-DETR (Ours) §	**22.2**	**35.0**	**22.8**

TABLE 5.4: Ablation study on region-level detection *vs.* image-level detection. "R" denotes region-level detection. "I" denotes image-level detection.

Method	aligned network	R/I	Novel mAP@0.5				
			1	2	3	5	10
FsDetView [64]		R	24.2	35.3	42.2	49.1	57.4
FsDetView + Deform. Trans.	✓	R	**28.0**	36.3	41.8	48.9	57.4
Meta-DETR *w/o* CAM	✓	I	27.2	**42.1**	**50.5**	**52.9**	**59.3**

5.4.3.2 MS COCO

Table 5.3 presents experimental results on MS COCO. It can be seen that, although MS COCO is much more challenging than Pascal VOC with higher complexity like occlusions and large scale variations, Meta-DETR still outperforms all existing methods under all setups by even larger margins. This can be attributed to *(i)* the complete circumvention of even more inaccurate region proposals for novel classes (See Fig. 5.2(a)) caused by the higher complexity of MS COCO, and *(ii)* the effective exploitation of the correlations among more classes in MS COCO. In addition, Meta-DETR performs exceptionally well compared with other region-based methods under the stricter metric $AP_{0.75}$, which implies that our proposed Meta-DETR can effectively lift the constraint of inaccurate region proposals and produce more accurate few-shot object detection.

5.4.4 Ablation Study

We conduct comprehensive ablation experiments to verify the effectiveness of our design choices. Experimental results are averaged over 10 runs with different randomly sampled support datasets on the first class split of Pascal VOC.

Region-Level Detection *vs.* Image-Level Detection. From Table 5.1 and Table 5.3, we can find that fine-tuning Deformable DETR (Deformable-DETR+ft-full) generally outperforms fine-tuning Faster R-CNN (FRCN+ft-full), especially in the MS COCO dataset, where it is much harder to obtain accurate region proposals for novel classes due to higher complexity (see Fig. 5.2(a)). This observation aligns well with our insight that region-based detection frameworks tend to suffer

TABLE 5.5: Ablation study on the impact of Correlational Aggregation Module (CAM). "R" denotes region-level detection. "I" denotes image-level detection. "C" denotes the number of support classes to aggregate simultaneously, which can only be 1 without the proposed CAM.

Detection Framework	R/I	Correlational Aggr. Module (CAM)	C	Novel mAP@0.5 1	2	3	5	10
Meta-DETR	I		1	27.2	42.1	50.5	52.9	59.3
		✓	1	30.3	44.0	52.1	55.7	**62.0**
		✓	5	**35.1**	**49.0**	**53.2**	**57.4**	**62.0**
FsDetView [64]	R		1	24.2	35.3	42.2	49.1	57.4
		✓	5	**30.1**	**41.1**	**45.2**	**51.4**	**57.5**

from inaccurate regional proposals for novel classes. To further verify the superiority of image-level few-shot object detection, we adopt FsDetView [64], a state-of-the-art meta-learning-based few-shot detector built on top of Faster R-CNN, as a solid baseline to compare with our method. For a fair comparison, we add deformable transformers to FsDetView (denoted as FsDetView + Deform. Trans.) to rule out the performance difference brought by the transformer architecture. Furthermore, we replace our proposed CAM in Meta-DETR with the feature aggregation module proposed in FsDetView (denoted as Meta-DETR *w/o* CAM). As shown in Table 5.4, even with aligned network architecture and aggregation scheme, Meta-DETR *w/o* CAM still outperforms FsDetView + Deform. Trans. under most setups. The results validate the superiority of solving few-shot object detection at image level.

Impact of Correlational Aggregation Module (CAM). As shown in Table 5.5, when incorporating CAM into our model, even if we keep the number of support classes for simultaneous aggregation (C) as 1, CAM can still boost few-shot detection performance under all settings. This demonstrates CAM's strong capacity in aggregating query and support information even without the leverage of inter-class correlation. When multiple support classes are available ($C \geq 2$), CAM can further exploit their inter-class correlation to boost few-shot detection performance under lower-shot (≤ 5) settings, especially under 1-shot (+ 4.8% mAP) and 2-shot (+ 5.0% mAP), which shows the benefit of inter-class correlational meta-learning. No clear performance gain is observed for 10-shot, which implies that, when more training samples are available, the detector can already recognize novel

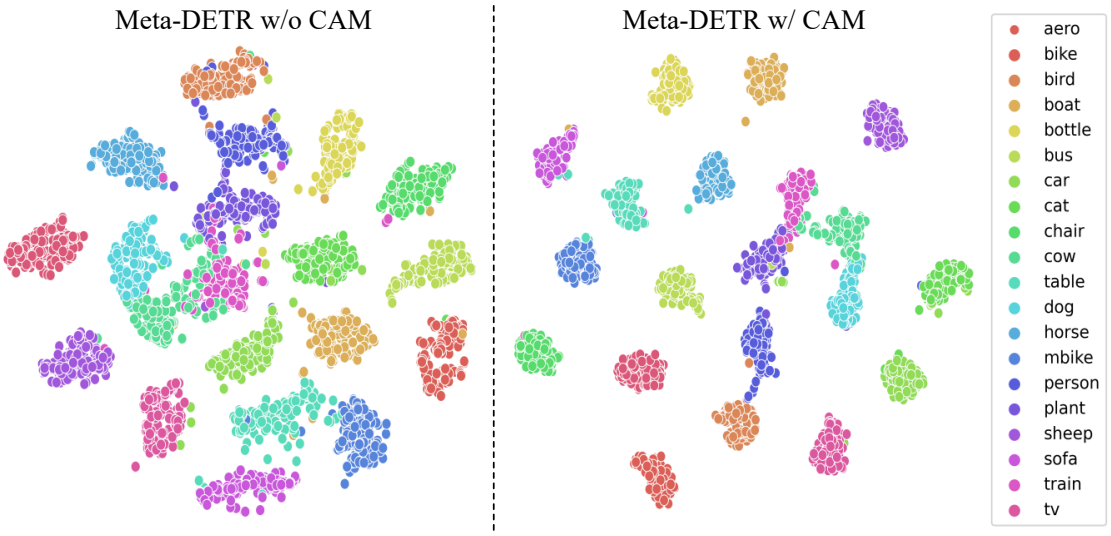


FIGURE 5.5: t-SNE visualization of objects learned in the feature space with and without our designed Correlational Aggregation Module (CAM). Results are obtained on Pascal VOC class split 1 under the 2-shot setup.

classes and differentiate them from similar classes without explicitly modeling the inter-class correlation. We also apply our designed CAM to the commonly used region-based meta-detector FsDetView [64] and report the results in Table 5.5. Its steady performance gain demonstrates that CAM and the proposed inter-class correaltional meta-learning strategy can also benefit region-level few-shot object detection.

To understand how CAM functions to improve detection accuracy, we visualize the objects from different classes in the feature space learned with and without the proposed CAM with t-SNE [127]. As shown in Fig. 5.5, with CAM included to perform inter-class correlational meta-learning, object classes are better separated from each other, which affirms our motivation of leveraging inter-class correlation to reduce mis-classification among similar classes. To further verify our claim that CAM effectively reduces mis-classification among similar classes, we select two pairs of similar classes (motorbike *vs.* bike and cow *vs.* horse) and plot their confusion matrices in Table 5.6. We can observe that CAM indeed reduces the mis-classification by large margins with the exploitation of inter-class correlation. We also observe fewer missed predictions, which shows that the effective leverage of inter-class correlations also facilitates generalization to detect previously missed cases.

TABLE 5.6: Confusion matrices of similar class pairs predicted with and without the proposed Correlational Aggregation Module. Results are obtained on Pascal VOC class split 1 under the 2-shot setup. "GT" denotes ground truth label; "Pred" denotes predicted label.

Meta-DETR w/o CAM

GT \ Pred	missed	mbike	bike
mbike	89	247	33
bike	63	10	316

GT \ Pred	missed	cow	horse
cow	82	218	29
horse	36	32	327

Meta-DETR w/ CAM

GT \ Pred	missed	mbike	bike
mbike	67	286	16
bike	58	7	324

GT \ Pred	missed	cow	horse
cow	46	273	10
horse	25	23	347

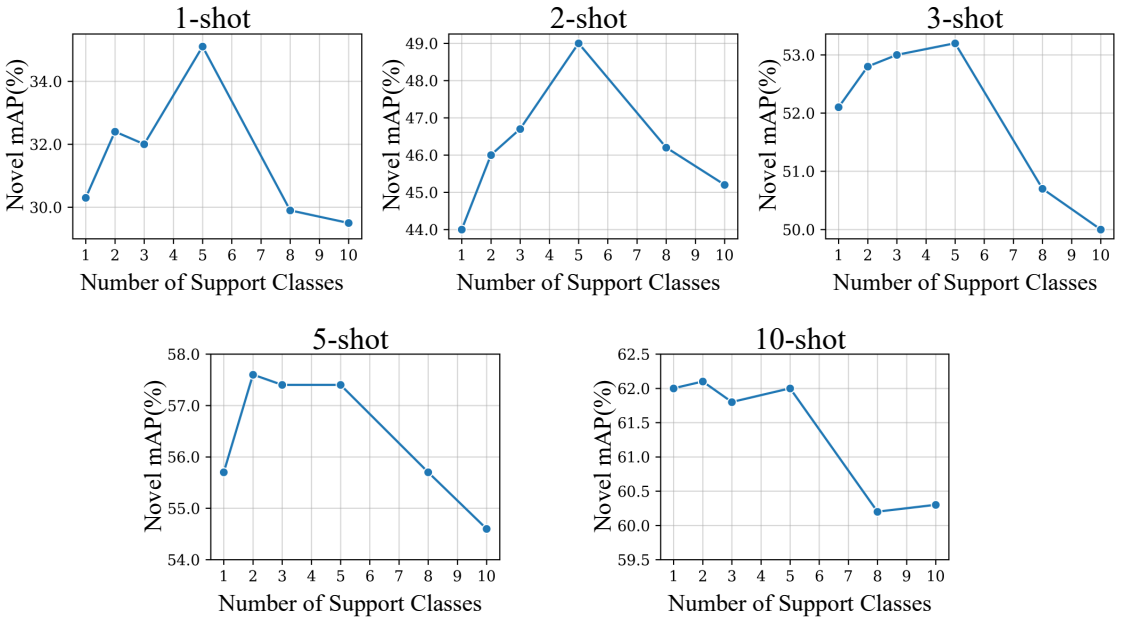


FIGURE 5.6: Ablation study on the number of support classes for simultaneous correlational aggregation under different few-shot setups. Results are averaged over 10 repeated runs on Pascal VOC class split 1.

Number of Classes for Correlational Aggregation. Meta-DETR receives a fixed number of support classes (C) and simultaneously aggregates them with query features to capture the inter-class correlation among different support classes. With $C \geq 2$, Meta-DETR exploits the inter-class correlation among different classes. Fig. 5.6 investigates the impact of the number of support classes for aggregation.

TABLE 5.7: Ablation study on the design choices of the attention mechanism in the proposed Correlational Aggregation Module (CAM).

(a) Apply Sigmoid	*(b)* Query Multiplication	*(c)* Modeling Background	Novel mAP@0.5 1	2	3	5	10
			29.8	44.8	51.2	54.8	59.6
		✓	31.2	46.1	52.5	56.2	61.5
✓	✓		32.6	45.6	51.3	56.1	60.9
✓	✓	✓	**35.1**	**49.0**	**53.2**	**57.4**	**62.0**

As the number of support classes C increases from 1 to 10, the lower-shot ($\leq$5) detection performance first improves and then drops, while 10-shot performance first saturates and then drops. This validates the effectiveness of leveraging inter-class correlation under lower-shot ($\leq$5) settings. The performance gain is considerable under extremely low-shots like 1-shot and 2-shot, indicating that it is highly beneficial to explore inter-class correlation when training samples are too scarce to model a novel class and differentiate it with other classes. We conjecture that the performance drop with a large number of support classes ($\geq$8) for correlational aggregation is due to the model's limited capacity to differentiate too many support classes at one go. Based on the results, we set our method's number of support classes C as 5 in all other experiments unless otherwise stated.

Design Choices for Correlational Aggregation Module (CAM). The proposed CAM's attention mechanism differs from the original DETR attention in three aspects: *(a)* applying a sigmoid function to attention's *Value* in feature matching, *(b)* multiplying attention's output with attention's *Query* in feature matching, and *(c)* explicitly modelling a prototype for the 'background' class. Among them, *(a)* and *(b)* are designed as a whole with *(a)* for generating 'filters' to remove query features that are irrelevant to the given support classes and *(b)* for applying the learned 'filters' to the query image features. And *(c)* enables Meta-DETR to better handle the 'no match' scenario where the query features do not match any of the support classes. We present ablation experiments in Table 5.7 that verify the effectiveness of the above three modifications.

Early Aggregation *vs.* Late Aggregation. The proposed CAM replaces one encoder layer in the transformer. As shown in Fig. 5.3, we place CAM ahead of the transformer encoder (as the first layer of the encoder). Table 5.8 studies the impact of the location of CAM in the transformer encoder. As shown, it is preferable to

TABLE 5.8: Ablation study on early aggregation *vs.* late aggregation.

CAM's Location @ Encoder Layers	Novel mAP@0.5 1	2	3	5	10
1	**35.1**	**49.0**	**53.2**	**57.4**	**62.0**
3	27.1	42.9	50.6	54.0	59.2
6	15.2	31.5	37.7	50.3	53.4

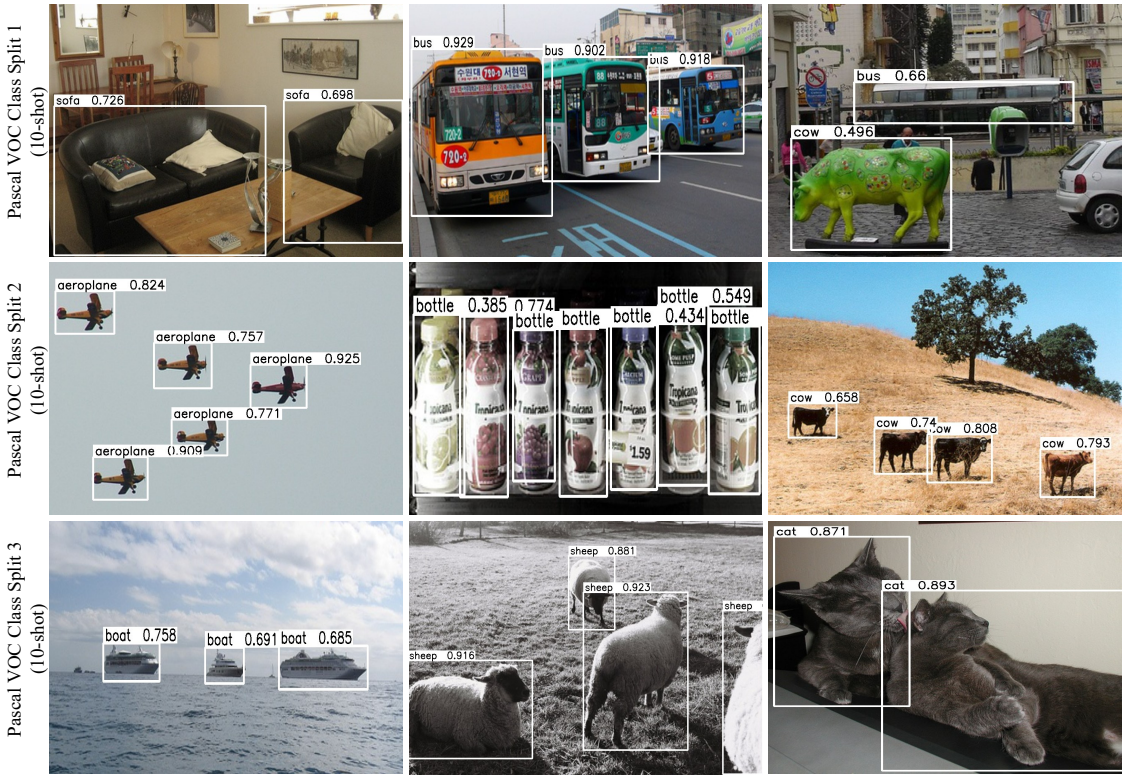


FIGURE 5.7: Visualization of Meta-DETR's 10-shot object detection results on various data setups of Pascal VOC. For simplicity, only detections of novel-class objects are illustrated. The qualitative experimental results show that Meta-DETR can detect novel objects effectively with very constrained training samples.

place CAM at the beginning stage of the transformer encoder for early aggregation, which also suggests the importance of learning a deep class-agnostic predictor.

5.4.5 Qualitative Results

In Fig. 5.7, we provide qualitative visualization of Meta-DETR's 10-shot object detection results on several sample images from their respective data setups. Note

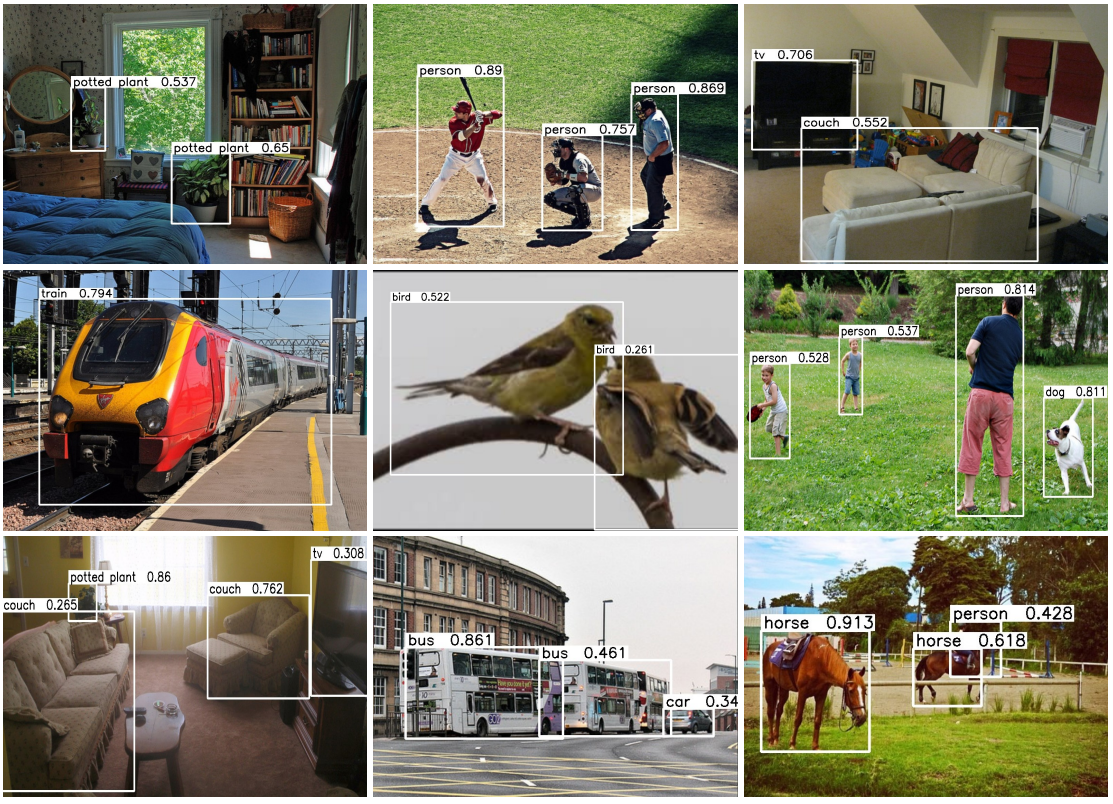


FIGURE 5.8: Visualization of Meta-DETR's 10-shot object detection results on MS COCO. For simplicity, only detections of novel-class objects are illustrated. The qualitative experimental results show that Meta-DETR can detect novel objects effectively with very constrained training samples.

that we show the detection of novel classes only since the focus of few-shot object detection is to detect objects of novel classes. We show detection results with confidence scores higher than 0.25 to filter out low-confidence predictions. It can be observed that the proposed Meta-DETR is able to detect novel objects effectively even with very limited training samples.

We also present qualitative visualization of Meta-DETR's 10-shot object detection results on MS COCO in Fig. 5.8. As Fig. 5.8 shows, even under the more challenging scenarios, Meta-DETR can also deliver generally satisfactory few-shot object detection performance.

FIGURE 5.9: Visualization of some failure cases of Meta-DETR's 10-shot object detection results. For simplicity, only detections of novel-class objects are illustrated. White boxes indicate true positives. Red solid boxes indicate false positives. Red dashed boxes indicate false negatives.

5.4.6 Typical Failure Cases

Fig. 5.9 illustrates typical failure cases of the proposed Meta-DETR. The most typical failure cases happen while multiple instances of novel objects are heavily clustered, largely due to the lack of supervision in such cases and the lack of a mechanism to discriminate objects' boundaries. Other typical failure cases include difficulty in detecting small objects as well as false negatives with less salient objects, which are also applicable in general object detectors.

TABLE 5.9: Few-shot instance segmentation performance on MS COCO for novel classes.

Shot	Method	Box						Mask					
		$AP_{0.5:0.95}$	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L	$AP_{0.5:0.95}$	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L
5	Mask-RCNN+ft-full [23]	1.3	3.0	1.1	0.3	1.1	2.4	1.3	2.7	1.1	0.3	0.6	2.2
	Meta R-CNN [27]	3.5	9.9	1.2	1.2	3.9	5.8	2.8	6.9	1.7	0.3	2.3	4.7
	Meta-DETR (Ours)	**15.3**	**24.9**	**15.4**	**1.5**	**12.8**	**26.0**	**8.1**	**16.8**	**7.1**	**0.9**	**5.6**	**13.7**
10	Mask-RCNN+ft-full [23]	2.5	5.7	1.9	2.0	2.7	3.9	1.9	4.7	1.3	0.2	1.4	3.2
	Meta R-CNN [27]	5.6	14.2	3.0	2.0	6.6	8.8	4.4	10.6	3.3	0.5	3.6	7.2
	Meta-DETR (Ours)	**19.8**	**31.3**	**20.4**	**4.5**	**17.4**	**30.5**	**10.1**	**20.8**	**8.7**	**1.7**	**7.6**	**15.8**

5.4.7 Extension to Few-Shot Instance Segmentation

The proposed Meta-DETR adopts a meta-learning framework which is generic and can be adapted to other downstream vision tasks beyond object detection. We validate this feature by examining how it can be extended to perform instance segmentation with simple modifications.

As described in [1], the original DETR can be extended to perform instance segmentation by adding a mask head on top of the decoder outputs. We similarly introduce an additional mask head over Meta-DETR to predict objects' masks for few-shot instance segmentation. The additional mask head takes the output of the transformer decoder and encoded image features as input and predicts a binary mask for each object query. It also follows the designed inter-class correlational meta-learning strategy for better generalization. To train Meta-DETR to perform few-shot instance segmentation, we first train it on the previously mentioned few-shot object detection tasks, and then freeze all the weights and train only the additional mask head for instance segmentation.

Experimental Results. We conduct experiments for few-shot instance segmentation on MS COCO under 5-shot and 10-shot setups. Similarly, the 20 classes shared with Pascal VOC are chosen as novel classes, and the remaining 60 classes are set as base classes. Note that AP for instance segmentation is evaluated with mask IoU. As shown in Table 5.9, Meta-DETR outperforms compared methods by large margins. The results demonstrate the superiority and universality of our Meta-DETR, which can extend to other instance-level few-shot learning tasks. Note that the compared Meta R-CNN [27] adopts region-level prediction together with the conventional class-by-class meta-learning via feature reweighting. The

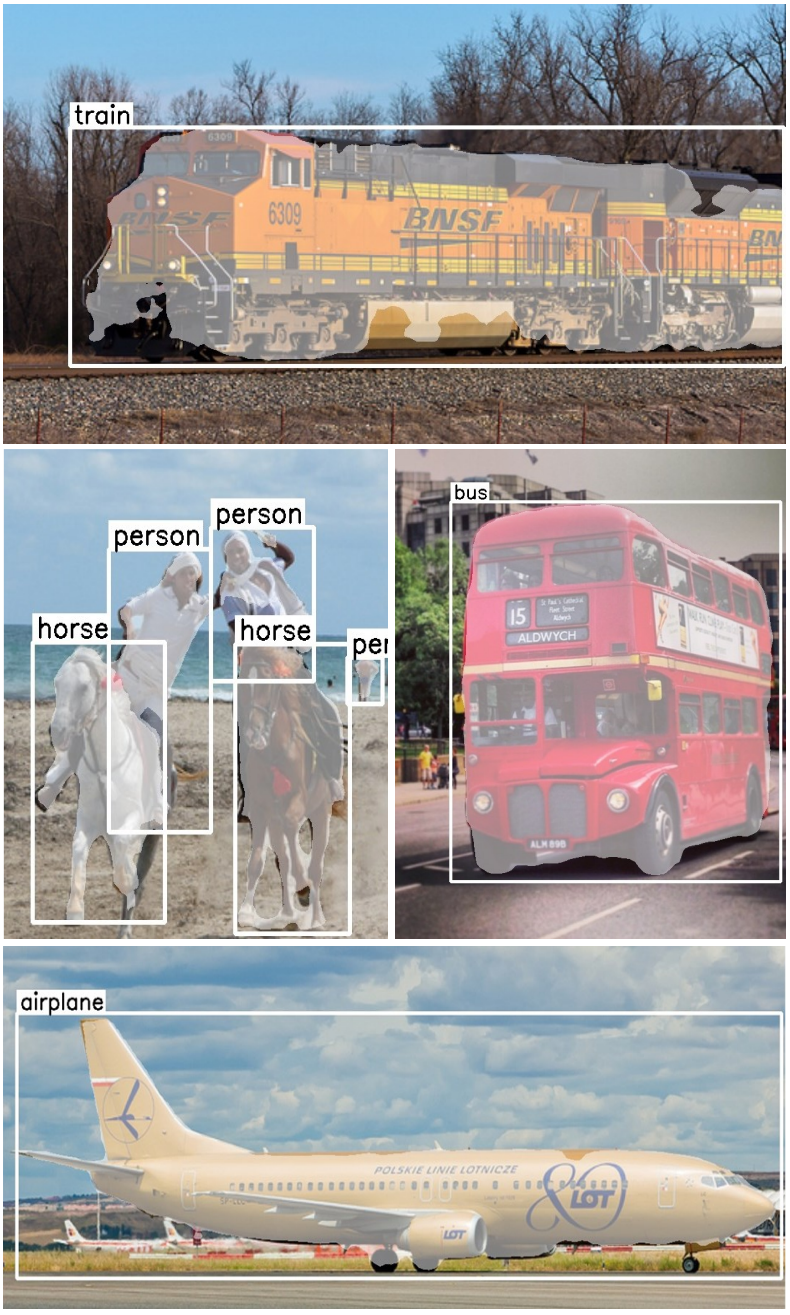


FIGURE 5.10: Visualization of Meta-DETR's 10-shot instance segmentation results on MS COCO. For simplicity, only detections and segmentations of novel-class objects are illustrated.

comparison between Meta R-CNN [27] and our proposed Meta-DETR verifies that the combination of the image-level prediction and the exploitation of inter-class correlation via correlational meta-learning can effectively benefit other instance-level few-shot learning tasks like few-shot instance segmentation. We also provide qualitative results for instance segmentation in Fig. 5.10.

5.5 Conclusion

This chapter studies object detection in a highly challenging setting – to detect objects of novel classes with only a few training samples available. Performing object detection under this constrained scenario is of great practical significance, as large-scale and annotated datasets are not always available for various specific applications due to expensive human labeling costs and difficulty in data acquisition. Most existing object detectors usually suffer from a catastrophic performance drop and fail to deliver decent detection accuracy.

To adapt to this constrained scenario, we propose a novel few-shot object detection framework in this chapter, namely Meta-DETR. The proposed Meta-DETR achieves *(i)* pure image-level prediction, which lifts the constraints caused by novel classes' inaccurate region proposals, and *(ii)* effective exploitation of categorical correlation via a inter-class correlational meta-learning strategy, which reduces misclassification and enhances generalization among similar or related classes. Despite its simplicity, Meta-DETR achieves state-of-the-art performance over multiple few-shot object detection setups, outperforming prior works by large margins. It can also be easily extended to other instance-level few-shot learning tasks. Being conceptually simple, powerful, and extendable, Meta-DETR is an excellent few-shot object detection framework for further research and application.

Chapter 6

IMFA: Towards Efficient Use of Multi-Scale Features in Transformer-Based Object Detectors [1]

6.1 Introduction

Detecting objects of vastly different scales has always been a major difficulty in object detection [4]. Fortunately, strong evidence [22, 25, 37, 38, 84, 128] shows that object detectors can significantly benefit from multi-scale features while dealing with large scale variation. For ConvNet-based object detectors like Faster R-CNN [21], Feature Pyramid Network (FPN) [22] and its variants [37, 104, 128–132] have become the go-to components for exploiting multi-scale features.

Different from those ConvNet-based object detectors [21, 26, 28, 29, 31, 34, 35, 37, 70, 104], the recently proposed DEtection TRansformer (DETR) [1] has established a new object detection paradigm based on the Transformer [36] architecture. Such Transformer-based object detectors [1, 38–41, 84] remove a series of hand-crafted

[1] The work in this chapter has been submitted to and is currently under review at the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2023 (**CVPR 2023**).

components and achieve fully end-to-end object detection with superior performance. However, these detectors often struggle while dealing with multi-scale features, largely due to the poor efficiency of the attention mechanism in processing high-resolution feature maps. Concretely, to process a feature map with a spatial size of $H \times W$, ConvNet requires a computational cost of $O(HW)$, while the complexity of the attention mechanism in Transformer-based object detectors is $O(H^2W^2)$. To mitigate this issue, Deformable DETR [38] and Sparse DETR [68] replace the original global dense attention with sparse attention. SMCA-DETR [84] restricts most Transformer encoder layers to be scale-specific, with only one encoder layer to integrate multi-scale features. However, as the number of tokens increases quadratically *w.r.t.* feature map size (typically 20x∼80x compared to single-scale), these methods [38, 68, 84] are still costly in computation and memory consumption, and rely on special operations like deformable attention [38] that introduces extra complexity for deployment. To the best of our knowledge, there is yet no generic approach that can efficiently exploit multi-scale features for Transformer-based object detectors.

In many practical applications such as autonomous driving, robotics, and augmented reality (AR), object detection needs to be performed efficiently on computationally limited devices. However, the lack of a general paradigm for efficient use of multi-scale features in Transformer-based object detectors hinders their application in those complex real-world scenarios, in which the exploitation of multi-scale features is necessary to deliver satisfactory detection accuracy. In this chapter, we aim to bridge this gap.

This chapter presents *Iterative Multi-scale Feature Aggregation (IMFA)*, a concise and effective technique that can serve as a generic paradigm for efficient use of multi-scale features in Transformer-based object detectors. The motivation comes from two key observations: *(i)* the computation of high-resolution features is highly redundant as the background usually occupies most of the image space, thus only a small portion of high-resolution features are useful to object detection; *(ii)* unlike ConvNet, the Transformer's attention mechanism does not require grid-shaped feature maps, which offers the feasibility of aggregating multi-scale features only from some specific regions that are likely to contain objects of interest. The two observations motivate us to sparsely sample multi-scale features from just a few

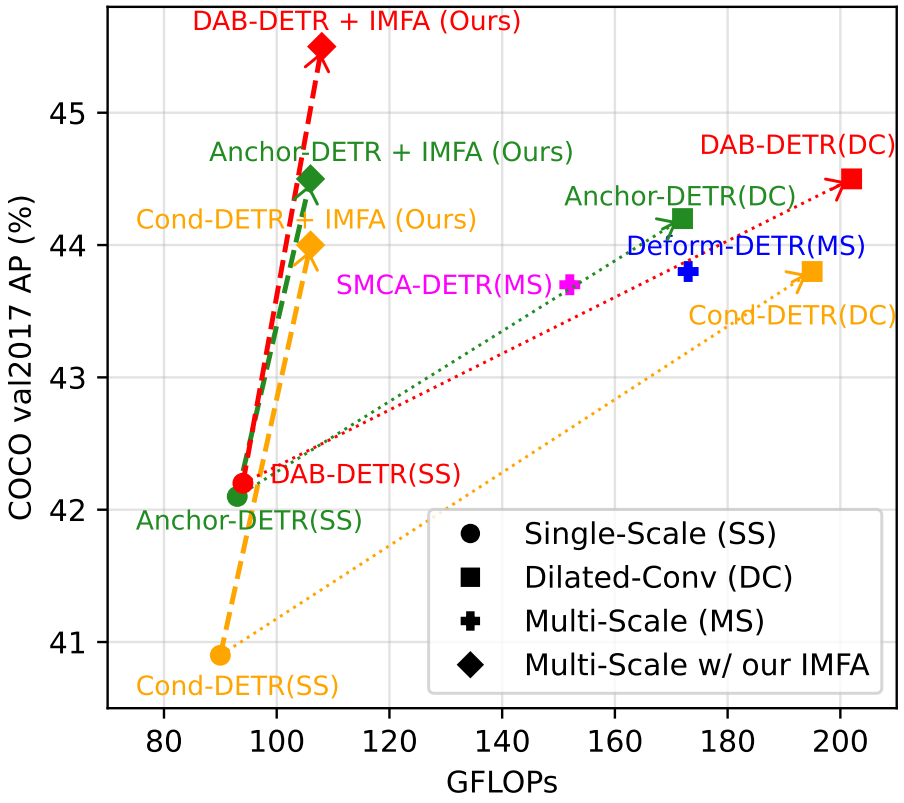


FIGURE 6.1: The proposed Iterative Multi-scale Feature Aggregation (IMFA) is a generic approach for efficient use of multi-scale features in Transformer-based object detectors. It significantly boosts detection accuracy on multiple object detectors at a minimal cost of additional computational overhead. All models adopt ResNet-50 as the backbone network. Best viewed in color.

informative locations and then aggregate them with encoded image features in an iterative manner.

Concretely, IMFA consists of two novel designs in the Transformer-based detection pipelines. *First*, IMFA rearranges the encoder-decoder pipeline so that each encoder layer is immediately connected to its corresponding decoder layer. This design enables iterative update of encoded image features along with refined detection predictions. *Second*, IMFA sparsely samples multi-scale features from the feature pyramid generated by the backbone, with the sampling process guided by previous detection predictions. Specifically, motivated by the spatial redundancy of high-resolution features, IMFA only focuses on a few promising regions with high likelihood of object occurrence based on prior predictions. Furthermore, inspired by the significance of objects' keypoints for recognition and localization [42, 101–103], IMFA first searches several keypoints within each promising region, and then

samples useful features around these keypoints at adaptively selected scales. The sampled features are finally fed to the subsequent encoder layer along with the encoded image features encoded by the previous layer. With the two new designs, the proposed IMFA aggregates only the most crucial multi-scale features from those informative locations. Since the number of the aggregated features is small, IMFA introduces minimal computational overhead while consistently improving the detection performance of Transformer-based object detectors. It is noteworthy that IMFA is a generic paradigm for efficient use of multi-scale features, *i.e.*, it can be easily integrated with many Transformer-based object detectors with consistent performance boost, as shown in Fig. 6.1.

The contributions of the work in this chapter are summarized as follows. *First*, we propose a novel Transformer-based detection pipeline, where encoded features can be iteratively updated along with refined detection predictions. This new pipeline allows to leverage intermediate object detection predictions as guidance for robust and efficient multi-scale feature encoding. *Second*, we propose a sparse sampling strategy for multi-scale features, which first identifies several promising regions under the guidance of prior detections, then searches several keypoints within each promising region, and finally samples their features at adaptively selected scales. We demonstrate that such sparse multi-scale features can significantly benefit object detection. *Third*, based on the two contributions above, we propose *Iterative Multi-scale Feature Aggregation (IMFA)* – a simple and generic paradigm that enables efficient use of multi-scale features in Transformer-based object detectors. The proposed IMFA consistently boosts detection performance on multiple object detectors, yet remains computationally efficient. To the best of our knowledge, this is the first work that investigates a generic approach for exploiting multi-scale features efficiently in Transformer-based object detectors.

6.2 Preliminaries

6.2.1 Multi-Scale Features for Object Detection

Modern object detectors can be divided into two categories: ConvNet-based object detectors and Transformer-based object detectors. ConvNet-based object detectors [21, 24, 27, 54, 61, 63, 64, 66, 89, 100, 115, 116, 118–120, 124, 126, 133–137]

have achieved promising results on various object detection challenges. However, these methods detect objects by defining surrogate regression and classification tasks, which rely on many hand-crafted components like anchors. Thus the detection pipelines of these ConvNet-based detectors are complex, hyper-parameter-intensive, and not fully end-to-end, leading to sub-optimal performance. Unlike ConvNet-based detectors, the recently proposed DETR [1] has revolutionized the paradigm for object detection using a Transformer [36] encoder-decoder architecture. Supervised by a set-based global loss, DETR [1] achieves competitive object detection results with a simple pipeline without the need for those hand-crafted components. Inspired by DETR [1], many Transformer-based object detectors [38–43, 67, 68, 84, 85, 109, 138–145] are proposed and achieve state-of-the-art detection accuracy as well as fast convergence.

One major difficulty in object detection is to effectively represent objects at vastly distinct scales. This is especially crucial for detecting small objects in images. Modern ConvNet-based object detectors [22, 26, 31, 32, 35, 37, 56, 104] usually exploit multi-scale features in the form of feature pyramids to accommodate this. As the pioneering work, Feature Pyramid Network (FPN) [22] adopts a top-down path to aggregate multi-scale features for generating feature pyramids. The paradigm of exploiting multi-scale features with FPN is further extended by many works [37, 104, 128–130, 132], and FPN has become a fundamental component for ConvNet-based object detectors. However, as feature pyramids require computation on high-resolution feature maps, FPN and its variants also introduce substantial computational overhead.

Multi-scale features are also helpful for Transformer-based detectors. However, due to the inefficiency of the Transformer's attention mechanism to process high-resolution feature maps, it requires special modifications to reduce the computational complexity to a feasible level. Concretely, Deformable DETR [38] proposes deformable attention, which reduces the complexity via key sparsification in the attention module. SMCA-DETR [84] uses only one multi-scale attention encoder layer while restricting other layers to be scale-specific. CF-DETR [140] embeds the Transformer encoder into an FPN [22] to produce feature pyramids, and extracts multi-scale features with RoIAlign [23]. These methods enable the use of multi-scale features in Transformer-based detectors, but introduce huge computational

overhead, require large-memory GPUs for training and inference, and rely on special modules like deformable attention [38] or RoIAlign [23]. To the best of our knowledge, there is no generic approach to efficiently leverage multi-scale features for Transformer-based detectors so far.

6.2.2 Spatial Redundancy and Sparse Features

Not all features are equally important. In most cases, only a small portion of features are crucial for recognition. With this motivation, several works [38, 67, 68, 146–149] perform sparse operations over feature maps to avoid computation at less informative locations. Specifically, in object detection, AutoFocus [150] first predicts and crops regions at coarse scales, and then makes final predictions on those regions at a higher resolution. PnP-DETR [67] and Sparse DETR [68] adaptively allocate encoding operations to informative feature tokens. One similar work to our proposed IMFA is QueryDet [69], which first coarsely predicts over low-resolution features, and then sparsely exploits multi-scale features based on the coarse predictions to generate the final detection results, thus improving inference speed. However, unlike our proposed IMFA, QueryDet [69] is designed for single-stage ConvNet-based detectors with FPN [22], and it only accelerates the inference procedure.

Our proposed IMFA is also motivated by the spatial redundancy in high-resolution features. IMFA defines a generic paradigm to efficiently exploit multi-scale features by iteratively and sparsely aggregating multi-scale features from just a few informative locations. As a result, IMFA exploits sparse yet still highly useful multi-scale features, which only adds slight computation cost while consistently improving the detection accuracy of Transformer-based object detectors.

6.3 A Revisit of DETR

Since our proposed method is developed on top of the recently proposed Transformer-based object detectors, we first briefly review the detection pipeline of Transformer-based object detectors [1, 38–41, 84], taking the pioneering work DETR [1] as an example.

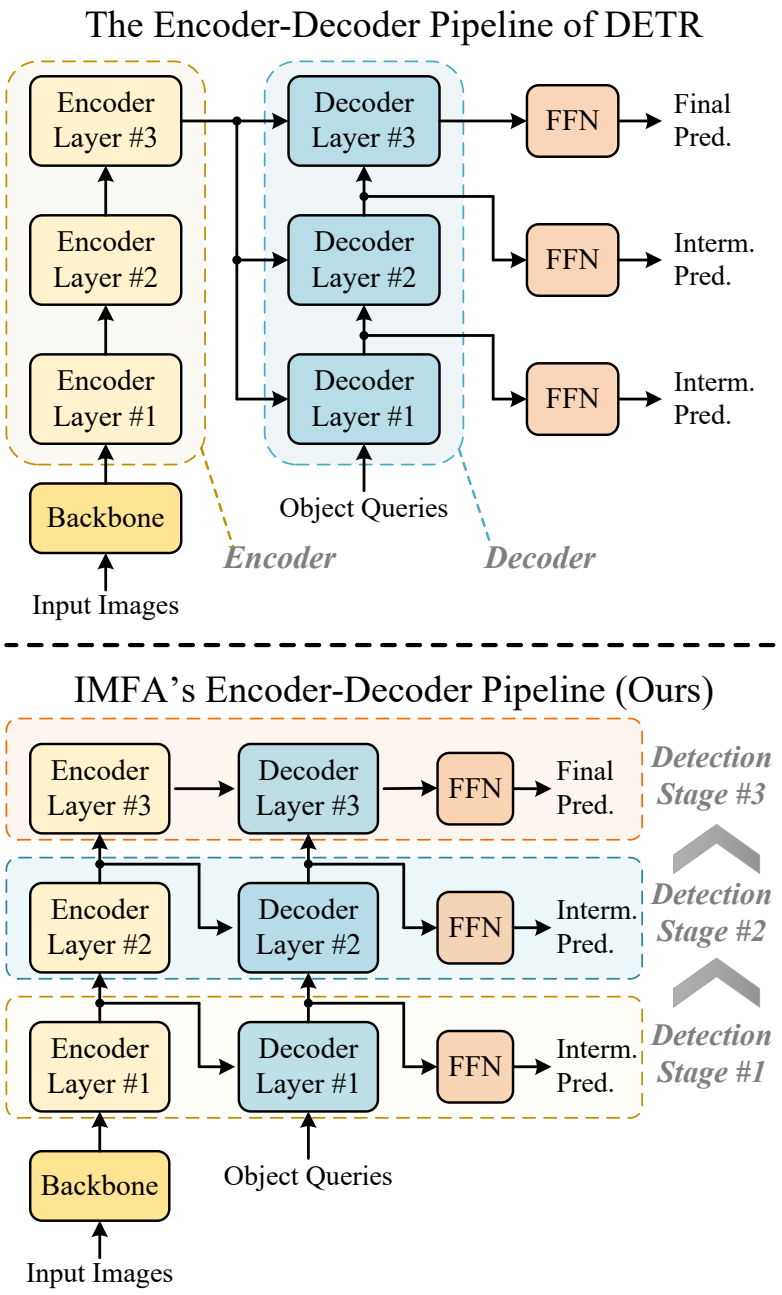


FIGURE 6.2: **Upper:** Most existing Transformer-based object detectors employ stacked Transformer encoder layers to obtain a fixed set of encoded image features, which are fed to each Transformer decoder layer to interact with object queries. Only object queries and their corresponding detection predictions are iteratively updated. **Lower:** IMFA rearranges the Transformer encoder-decoder pipeline into multiple stacked detection stages. Each detection stage is composed of an encoder layer, a decoder layer, and a feed-forward network (FFN), in which encoded features, object queries, and detection predictions can all be iteratively updated during the detection refinement process. Only three encoder and decoder layers are presented for illustration only.

DETR [1] formulates object detection as a direct set prediction problem and uses a Transformer [36] encoder-decoder architecture to solve it. Given an image $\mathbf{I} \in \mathbb{R}^{H_0 \times W_0 \times 3}$, the backbone network generates its feature maps, which are further fed to the Transformer encoder to produce the encoded image features $\mathbf{F} \in \mathbb{R}^{HW \times d}$, where d denotes the feature dimension, and H_0, W_0 and H, W are the spatial sizes of the input image and its feature maps, respectively. Then, the encoded features are fed to the Transformer decoder to interact with a set of object queries representing objects at different spatial locations. The object queries are finally used to produce final detection predictions with a feed-forward network (FFN). The entire detection pipeline is supervised by a set-based global loss with bipartite matching.

Specifically, both the Transformer encoder and decoder are composed of multiple layers. As shown in Fig. 6.2 (upper), existing Transformer-based object detectors [1, 38–40, 42, 84] usually process the input image features with a stack of encoder layers and obtain a fixed set of encoded features, which are further fed to the Transformer decoder layers to update the detection results iteratively. Differently, as illustrated in Fig. 6.2 (lower), one major difference introduced by IMFA is that it rearranges the encoder-decoder pipeline into multiple stacked detection stages, so that encoded features can also be iteratively updated along with detection predictions. This design modification lays the foundation for efficient use of multi-scale features guided by prior detection results, which is to be detailed in the next section.

6.4 Methodology

This section provides a detailed description of the proposed *Iterative Multi-scale Feature Aggregation (IMFA)*, which can serve as a generic paradigm for efficient use of multi-scale features in Transformer-based object detectors such as DETR [1] and its extensions [39–41].

6.4.1 Overview

Fig. 6.3 illustrates the detection pipeline of the proposed IMFA. For computational efficiency, IMFA exploits multi-scale features with dual-sparsity: *(i)* it samples multi-scale features from just a few promising regions with high likelihood of object occurrence as guided by prior detection predictions; *(ii)* for each promising region, it only samples features from several keypoints with the most informative features at adaptively selected scales. The dual-sparsity is achieved with two novel designs, which are to be described in detail in the following subsections.

6.4.2 Iterative Update of Encoded Features

The iterative update of encoded image features is the basis for IMFA to exploit multi-scale features efficiently. As introduced in Section 6.3, most existing Transformer-based detectors use fixed encoded image features to make predictions. In order to guide the multi-scale sampling process with prior detections, IMFA rearranges the Transformer encoder-decoder pipeline, as shown in Fig. 6.2.

Specifically, instead of using stacked encoder layers to produce a fixed set of feature tokens at one go, IMFA rearranges the detection pipeline into several stacked *detection stages*. Each *detection stage* consists of an encoder layer, a decoder layer, and an FFN to produce detection predictions. This design lays the foundation for incorporating sparse multi-scale features dynamically under the guidance of prior detection predictions, which is detailed in Section 6.4.3. It is noteworthy that, according to the experiments in Section 6.5.5, this design alone (shown in Fig. 6.2 (right), without incorporating multi-scale features) brings no performance gain over the baseline model.

6.4.3 Sparse Multi-Scale Feature Sampling and Aggregation

Naively incorporating multi-scale features into the encoder leads to prohibitive computational complexity, as the number of feature tokens from all scales is too large to be processed by the attention mechanism. This motivates us to exploit only the most informative multi-scale features.

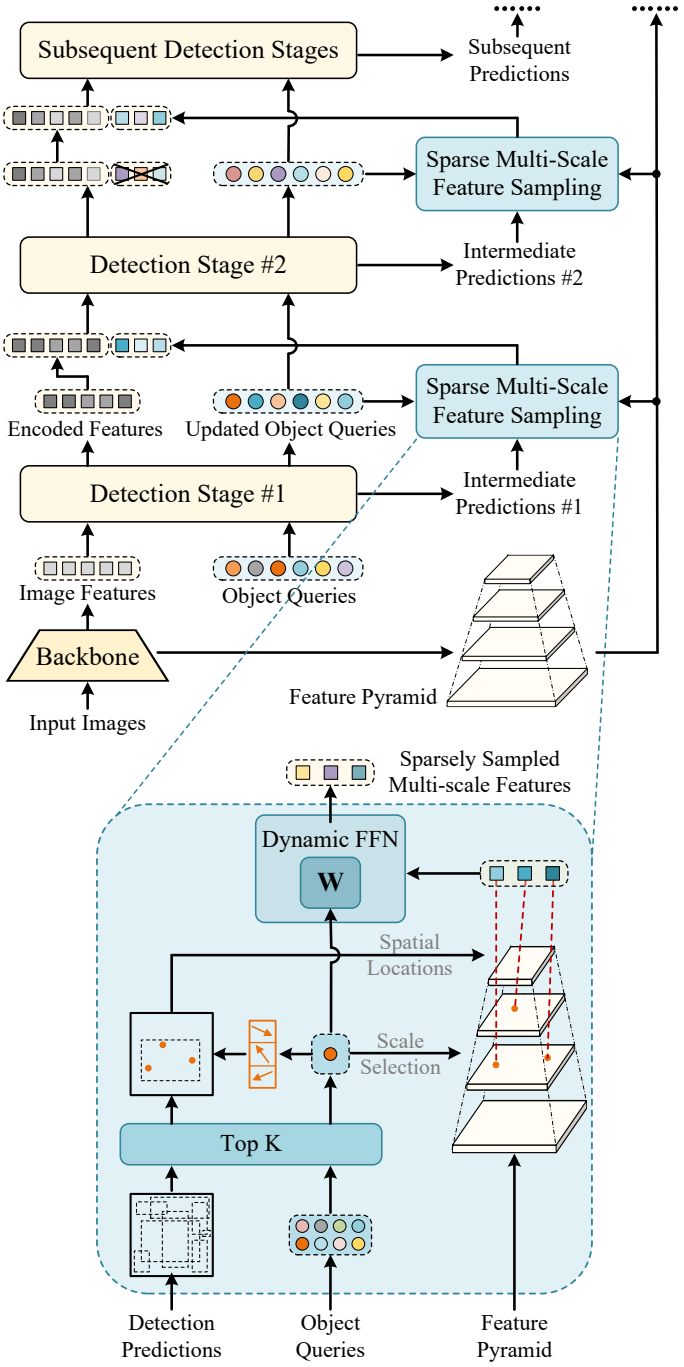


FIGURE 6.3: The detection pipeline of Iterative Multi-scale Feature Aggregation (IMFA). IMFA adopts the pipeline in Fig. 6.2 (lower) with multiple stacked detection stages, which enables the iterative update of encoded features. On this basis, IMFA performs sparse multi-scale feature sampling under the guidance of prior detection predictions. Specifically, it focuses on a few promising regions guided by prior detection predictions, then searches for several keypoints within each promising region, and finally samples features around these keypoints at adaptively selected scales. IMFA also adopts a Dynamic FFN to enhance the representation capacity of sparsely sampled multi-scale features by incorporating semantics from their corresponding object queries. The sampled features are fed into the subsequent detection stages along with encoded features for refined detection. Only the first two detection stages are presented for simplicity.

On the basis of Section 6.4.2, IMFA further performs sparse multi-scale feature sampling using prior detection predictions as guidance, as illustrated in Fig. 6.3. Specifically, IMFA first identifies a few promising regions with high likelihood of object occurrence. Then, it searches for several representative and informative keypoints within each promising region and samples their features at adaptively selected scales. Finally, the sampled features are fed to the subsequent encoder layers to aggregate with image features to produce refined detection predictions.

Identifying Promising Regions Based on Prior Predictions. In most cases, objects are sparsely distributed across images [6, 69, 150], which motivates us to exploit only the multi-scale features related to these objects. An intuitive solution is to guide the sampling process with the high-confidence detection predictions from the previous detection stage. Concretely, as shown in Fig. 6.3, for each detection stage except the first stage, we select K predictions with the highest classification confidence scores from the previous detection stage as the promising regions. Here, $K = N \times r$, with N denoting the number of object queries and r denoting IMFA's sampling ratio. Formally, we denote the selected box predictions and their corresponding object queries as $\{(\mathbf{B}_1, \mathbf{Q}_1), (\mathbf{B}_2, \mathbf{Q}_2), ..., (\mathbf{B}_K, \mathbf{Q}_K)\}$. The multi-scale features are then sampled within these promising regions, which is to be introduced in detail later. Since Transformer-based object detectors [1, 39–41, 84] already employ a sparse set (typically 100∼300) of object queries to represent different objects, the promising regions sampled by IMFA remain sparse for efficient computation.

Sampling Scale-Adaptive Features from Representative Keypoints. IMFA directly samples multi-scale features from the feature pyramid that is generated from the backbone (C2-C5 from ResNet in our experiments). However, even the sparsely sampled promising regions still contain a substantial amount of feature tokens at high-resolution feature scales. To further sparsify the sampled multi-scale features, IMFA searches a small number of representative keypoints within each promising region and samples their corresponding features at adaptively selected scales.

As illustrated in Fig. 6.3, for each promising region, IMFA first uses its corresponding object query to predict M keypoint locations within the region, which can be

formulated as:

$$\{P_{ij}\}_{j=1}^{M} = \text{MLP}(\mathbf{Q}_i) \quad \text{for } i = 1, 2, ..., K \ , \tag{6.1}$$

where i and j index the queries and keypoints, respectively, and each keypoint $P_{ij} = (x_{ij}, y_{ij})$ lies within its corresponding box prediction $\mathbf{B}_i$. Then, IMFA samples each keypoint's features from the feature pyramid at all scales via bilinear interpolation, obtaining a set of features $\{\mathbf{F}_{ij}^{s}\}_{s=1}^{S}$, where S is the number of feature scales. Finally, to emphasize the distinct significance of different feature scales for each keypoint, we propose to perform adaptive scale selection by predicting scale-specific weights for each keypoint and obtaining scale-adaptive features through weighted summation:

$$\mathbf{F}_{ij} = \textstyle\sum_s \alpha_{ij}^{s} \mathbf{F}_{ij}^{s} \qquad \{\alpha_{ij}^{s}\}_{s=1}^{S} = \text{Softmax}(\gamma_j(\mathbf{Q}_i)), \tag{6.2}$$

where the scale-selection weights α are generated by a linear projection γ_j followed by a Softmax function, so that $\sum_s \alpha_{ij}^{s} = 1$. In this way, IMFA only samples the most crucial and informative features, producing a set of sparse yet still highly informative multi-scale features for each promising region. Additionally, to further strengthen the representation capacity of the sampled multi-scale features, we feed the sampled features into a Dynamic Feed-Forward Network (Dynamic FFN) to incorporate the semantics from their corresponding object queries via dynamic weighting [108], where FFN's weights are dynamically generated by the object queries. It can be formulated as:

$$\mathbf{F}'_{ij} = \text{MLP}_{\mathbf{W}_i}(\mathbf{F}_{ij}) \qquad \mathbf{W}_i = \psi(\mathbf{Q}_i). \tag{6.3}$$

Here, for each object query $\mathbf{Q}_i$, the dynamic weight $\mathbf{W}_i$ is obtained by a linear projection ψ of $\mathbf{Q}_i$. Then, $\mathbf{W}_i$ is applied to the scale-adaptive features $\mathbf{F}_{ij}$ to generate the final sampled features $\mathbf{F}'_{ij}$ with enhanced semantics. These sampled features, along with their positional embeddings obtained based on their keypoint locations, are further fed to the subsequent detection stage for aggregation.

Iterative Aggregation of Multi-Scale Features. To leverage the sampled multi-scale features for refined object detection, the sampled features and the encoded image features are fed into the subsequent encoder layer for aggregation using the attention mechanism. This is analogous to the top-down path created by FPN-like architectures [22, 37, 104, 128–132] for enhancing the semantics of

low-level features. To avoid continuous growth of feature tokens and maintain efficiency, each detection stage does not inherit the multi-scale features that are generated from the previous stage, as illustrated in Fig. 6.3.

6.4.4 Network Optimization

As described in Section 6.4, all additional operations introduced by IMFA are fully differentiable, including the selection of top-K prior detection predictions, sparse feature sampling via bilinear interpolation, adaptive scale selection, Dynamic FFN, and iterative feature aggregation. Thus, the proposed IMFA can be trained end-to-end on top of the corresponding baselines [1, 39–41].

Besides, IMFA requires no additional training objective. In other words, IMFA is learned purely from the supervision signals of the corresponding baselines' detection-related losses.

6.5 Experiments

6.5.1 Dataset and Evaluation Metrics

We follow prior works [1, 39–41] and perform experiments on the MS COCO 2017 dataset [6]. We use ∼117k images in `train2017` for training and 5k images in `val2017` for evaluation. We adopt MS COCO's standard evaluation metrics for performance evaluation.

6.5.2 Implementation Details

As the proposed IMFA defines a generic paradigm, we mainly conduct experiments with DAB-DETR [41] – one of the state-of-the-art Transformer-based object detectors with open-sourced implementation. We also integrate IMFA with DETR [1], Conditional DETR [39], and Anchor DETR [40], to demonstrate the generality of our proposed method.

FIGURE 6.4: Visualization of IMFA's sampling locations and their adaptively selected feature scales. The searched sampling points mostly fall around the objects of interest, many of which are highly representative points with rich semantics, such as objects' extremities. Besides, IMFA adaptively selects appropriate feature scales for each sampling point, generating sparse yet informative scale-adaptive features for refined detection predictions. Best viewed in color.

For IMFA-related hyper-parameters, we set the sampling ratio r at 20% and the keypoint number M at 8 by default. Other model-related setups align with their corresponding baselines [1, 39–41]. We use ImageNet-pretrained [46] ResNet [76] as backbone networks, and conduct training with $4 \times$ Nvidia V100 GPUs using the AdamW optimizer [106, 107]. The batch size is set to 16 for training. The initial learning rate is $1{\times}10^{-5}$ for the backbone networks and $1{\times}10^{-4}$ for the Transformer

architectures, along with a weight decay of 1×10^{-4}. Models are trained for 50 epochs, with the learning rate decayed at the 40^{th} epoch by 0.1. The same data augmentation scheme used in [1, 38–42] is adopted, which includes random crop, random resize, and horizontal flip, with images' longest sides less than 1333 pixels and the shortest sides larger than 480 pixels.

6.5.3 Visualization and Analysis

Fig. 6.4 presents visualizations of IMFA's sampling locations and their feature scales. It can be observed that the sampling locations mostly fall around the target objects, and typically at representative locations, such as object extremities. This proves the effectiveness of IMFA in searching sparse yet highly informative locations in the feature sampling process. Besides, it is noteworthy that IMFA tends to focus on higher-resolution features for small objects and lower-resolution features for large objects, which is intuitive as the detection of small objects relies more on finer details.

6.5.4 Experiment Results

Compatibility with Transformer-Based Object Detectors. We first show the generality of the proposed IMFA by integrating it with multiple Transformer-based object detectors, including DETR [1] and its variants Conditional DETR [39], Anchor DETR [40], and DAB-DETR [41]. As discussed in Section 6.1, these methods resort to higher-resolution backbones (denoted with 'DC') as an alternative, as it is computationally prohibitive for them to directly process multi-scale features. As shown in Table 6.1, using higher-resolution features improves the detection performance but adds a substantial computational cost ($+\sim$100 GFLOPs) as well as GPU memory consumption. On the other hand, the proposed IMFA consistently improves the detection performance by large margins across all metrics, especially on small objects (AP_S), yet only introduces a slight computational overhead ($+\sim$15 GFLOPs). The experimental results demonstrate IMFA's effectiveness and wide applicability.

TABLE 6.1: Compatibility with Transformer-based object detectors. IMFA boosts the performance of existing detectors by large margins at a slight computational cost. 'MS' denotes the use of multi-scale features. 'DC' denotes the use of high-resolution features with R50-DC5. ‡ denotes DETR [1] with 300 object queries and focal loss. Results are reported on MS COCO val 2017.

Method	MS	DC	#Epochs	#Params	FLOPs	GPU Mem	AP	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L
DETR-R50 [1] ‡			50	41M	86G	2.1 GB	34.9	55.5	36.0	14.4	37.2	54.5
DETR-R50 [1] ‡		✓	50	41M	187G	5.8 GB	36.7	57.6	38.2	15.4	39.8	**56.3**
DETR-R50 [1] + IMFA (Ours) ‡	✓		50	52M	105G	2.5 GB	**39.2**	**58.8**	**41.6**	**20.3**	**42.2**	55.4
Cond-DETR-R50 [39]			50	44M	90G	2.1 GB	40.9	61.8	43.3	20.8	44.6	59.2
Cond-DETR-R50 [39]		✓	50	44M	195G	5.8 GB	43.8	**64.4**	46.7	24.0	**47.6**	**60.7**
Cond-DETR-R50 [39] + IMFA (Ours)	✓		50	53M	106G	2.5 GB	**44.0**	64.2	**47.5**	**25.7**	46.8	59.8
Anchor-DETR-R50 [40]			50	37M	93G	2.1 GB	42.1	63.1	44.9	22.3	46.2	60.0
Anchor-DETR-R50 [40]		✓	50	37M	172G	3.6 GB	44.2	**64.7**	47.5	24.7	**48.2**	**60.6**
Anchor-DETR-R50 [40] + IMFA (Ours)	✓		50	46M	106G	2.4 GB	**44.5**	63.9	**47.7**	**26.4**	47.7	59.9
DAB-DETR-R50 [41]			50	44M	94G	2.1 GB	42.2	63.1	44.7	21.5	45.7	60.3
DAB-DETR-R50 [41]		✓	50	44M	202G	6.0 GB	44.5	**65.1**	47.7	25.3	48.2	**62.3**
DAB-DETR-R50 [41] + IMFA (Ours)	✓		50	53M	108G	2.5 GB	**45.5**	65.0	**49.3**	**27.3**	**48.3**	61.6

Comparison with State-of-the-Art Object Detectors. We integrate the proposed IMFA with DAB-DETR [41] to benchmark with other state-of-the-art single-stage Transformer-based detectors that utilize high-resolution or multi-scale features. We also include some popular two-stage detectors [21, 110, 151] for a comprehensive comparison. As shown in Table 6.2, our method can achieve comparable performance with the state-of-the-art methods, but with significantly less computational cost.

As shown in Table 6.3, our method can further benefit from stronger Vision Transformer (ViT) [152] backbones. With Swin-Transformer-Tiny (Swin-T) [152] as the backbone, `DAB-DETR-Swin-T+IMFA` significantly outperforms `DAB-DETR-R50+IMFA` and `DAB-DETR-R101+IMFA`, with comparable and notably less computational cost, respectively. The results demonstrate IMFA's excellent scalability.

6.5.5 Ablation Study

We conduct ablation studies with the strong baseline DAB-DETR [41] to validate the effectiveness of our proposed designs. Experiments are performed with ResNet-50 [76] as the backbone.

Effect of IMFA's Design Choices. IMFA introduces two novel designs: the iterative encoding described in Section 6.4.2 and the iterative multi-scale feature

TABLE 6.2: Comparison with state-of-the-art object detectors on MS COCO val 2017. Our proposed method achieves comparable performance with the state-of-the-art methods, but with significantly lower computation. 'MS' and 'DC' denote the use of multi-scale and high-resolution features, respectively.

Method	MS	DC	#Epochs	#Params	FLOPs	AP	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L
Faster-RCNN-FPN-R50 [21, 22]	✓		108	42M	180G	42.0	62.1	45.5	26.6	45.5	53.4
TSP-FCOS-FPN-R50 [110]	✓		36	52M	189G	43.1	62.3	47.0	26.6	46.8	55.9
TSP-RCNN-FPN-R50 [110]	✓		36	64M	188G	43.8	63.3	48.3	**28.6**	46.9	55.7
Sparse-RCNN-FPN-R50 [108]	✓		36	106M	166G	45.0	64.1	48.9	28.0	47.6	59.5
DETR-R50 [1]		✓	500	41M	187G	43.3	63.1	45.9	22.5	47.3	61.1
Deformable-DETR-R50 [38]	✓		50	40M	173G	43.8	62.6	47.7	26.4	47.1	58.0
Deformable-DETR-R50 [38] + Iter	✓		50	41M	173G	45.4	64.7	49.0	26.8	48.3	61.7
Efficient-DETR-R50 [151]	✓		36	32M	159G	44.2	62.2	48.0	28.4	47.5	56.6
Conditional-DETR-R50 [39]		✓	50	44M	195G	43.8	64.4	46.7	24.0	47.6	60.7
SMCA-DETR-R50 [84]	✓		50	40M	152G	43.7	63.6	47.2	24.2	47.0	60.4
YOLOS-DeiT-S [139]			150	28M	172G	37.6	57.6	39.2	15.9	40.2	57.3
Anchor-DETR-R50 [40]		✓	50	37M	172G	44.2	64.7	47.5	24.7	48.2	60.6
DAB-DETR-R50 [41]		✓	50	44M	202G	44.5	65.1	47.7	25.3	48.2	62.3
SAM-DETR-R50 [42]		✓	50	58M	210G	43.3	64.4	46.2	25.1	46.9	61.0
SAM-DETR-R50 [42] w/ SMCA [84]		✓	50	58M	210G	45.0	**65.4**	47.9	26.2	**49.0**	**63.3**
DAB-DETR-R50 [41] + IMFA (Ours)	✓		50	53M	**108G**	**45.5**	65.0	**49.3**	27.3	48.3	61.6
Faster-RCNN-FPN-R101 [21, 22]	✓		108	60M	246G	44.0	63.9	47.8	27.2	48.1	56.0
TSP-FCOS-FPN-R101 [110]	✓		36	70M	255G	44.4	63.8	48.2	27.7	48.6	57.3
TSP-RCNN-FPN-R101 [110]	✓		36	83M	254G	44.8	63.8	49.2	29.0	47.9	57.1
Sparse-RCNN-FPN-R101 [108]	✓		36	125M	242G	**46.2**	65.1	**50.4**	**29.5**	49.2	61.7
DETR-R101 [1]		✓	500	60M	253G	44.9	64.7	47.7	23.7	49.5	62.3
Conditional-DETR-R101 [39]		✓	50	63M	262G	45.0	65.5	48.4	26.1	48.9	62.8
SMCA-DETR-R101 [84]	✓		50	58M	218G	44.4	65.2	48.0	24.3	48.5	61.0
YOLOS-DeiT-B [139]			150	127M	538G	42.0	62.2	44.5	19.5	45.3	62.1
Anchor-DETR-R101 [40]		✓	50	56M	238G	45.1	65.7	48.8	25.8	49.4	61.6
DAB-DETR-R101 [41]		✓	50	63M	282G	45.8	**65.9**	49.3	27.0	**49.8**	**63.8**
DAB-DETR-R101 [41] + IMFA (Ours)	✓		50	72M	**174G**	**46.2**	**65.9**	50.1	27.2	**49.8**	62.4

TABLE 6.3: Comparison with state-of-the-art object detectors with Vision Transformer (ViT) backbones on MS COCO val 2017. 'MS' denotes the use of multi-scale features. '§' denotes two-stage Transformer-based object detector, with the encoder producing 'region proposals' to initialize object queries.

Method	MS	#Epochs	#Params	FLOPs	AP	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L
DAB-DETR-R50 [41] + IMFA (Ours)	✓	50	53M	108G	45.5	65.0	49.3	27.3	48.3	61.6
DAB-DETR-R101 [41] + IMFA (Ours)	✓	50	72M	174G	46.2	65.9	50.1	27.2	49.8	62.4
DAB-DETR-Swin-T [41] + IMFA (Ours)	✓	50	57M	114G	**47.0**	67.1	**50.6**	29.5	**49.7**	**63.3**
Deformable-DETR-Swin-T [38]	✓	50	40M	180G	45.7	65.3	49.9	26.9	49.4	61.2
YOLOS-DeiT-B [139]		150	127M	538G	42.0	62.2	44.5	19.5	45.3	62.1
Sparse-DETR-Swin-T [68] §	✓	50	41M	113G	46.8	**68.0**	**50.6**	**29.7**	**49.7**	**63.3**
ViDT-Swin-T [144]	✓	50	38M	100G	44.8	64.5	48.7	25.9	47.6	62.1

aggregation described in Section 6.4.3. As shown in Table 6.4, the iterative encoding alone even slightly degrades the baseline's performance. However, with the IMFA's sparsely sampled multi-scale features, our method significantly improves the detection performance of objects at all scales, especially at smaller scales. This proves that the multi-scale features sampled by IMFA are sparse yet highly effective for object detection. We also examine two crucial components in the sparse

TABLE 6.4: Ablation study on IMFA's design choices. Results are obtained on MS COCO val 2017.

Design Choice	#Params	FLOPs	AP	AP_S	AP_M	AP_L
DAB-DETR [41]	44M	94G	42.2	21.5	45.7	60.3
+ Iterative Encoding (Fig. 6.2 (right))	44M	94G	41.9	21.8	45.2	61.1
+ IMFA w/o Dynamic FFN	45M	105G	44.2	26.3	47.2	60.8
+ IMFA w/o Adaptive Scale Selection	53M	108G	44.7	26.4	47.6	61.5
+ **IMFA (Ours)**	53M	108G	**45.5**	**27.3**	**48.3**	**61.6**

multi-scale feature sampling process. Without Dynamic FFN, the performance drops, which validates that Dynamic FFN successfully fuses important information from the corresponding object queries. The performance also drops without adaptive scale selection. This indicates that the adaptive scale selection mechanism can focus on appropriate scales for different objects, generating scale-adaptive features that can benefit object detection effectively.

Effect of IMFA's Hyper-Parameters. IMFA introduces two hyper-parameters: the sampling ratio of prior detection predictions and object queries (r) as well as the keypoint number in each promising region (M). We conduct sensitivity analysis on each of them.

Table 6.5 shows the effect of different r values when M is fixed at 8. It can be observed that as r increases from 10% to 30%, the average precision (AP) first increases then decreases, while the computational cost keeps growing. An interesting trend is that the detection performance of small objects (AP_S) goes up with increasing r consistently. We conjecture that small objects rely more on the fine details in high-resolution features, so that they can benefit from increased number of promising regions used for multi-scale feature sampling. However, the overall performance drops when r is too large, which we conjecture is due to the increased difficulty in searching relevant features with overwhelming feature tokens involved. Based on the experimental results, we set the default value for r as 20% in our system.

To study the effect of the number of keypoints M, we conduct experiments by fixing r at 20% and report the results in Table 6.6. We can see a similar trend that the performance improves as M increases but then drops when M becomes too large. Therefore, we set M as 8 by default.

TABLE 6.5: Ablation study on the sampling ratio r of prior detection predictions. The keypoint number M within each promising region is set to 8 by default. Results are obtained on MS COCO val 2017.

r	#Params	FLOPs	AP	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L
10%	53M	103G	44.2	64.0	47.5	25.9	47.3	60.6
15%	53M	105G	44.8	64.2	48.2	26.5	47.7	60.1
20%	53M	108G	**45.5**	65.0	**49.3**	27.3	**48.3**	**61.6**
25%	53M	111G	45.3	**65.1**	49.0	27.9	47.9	61.1
30%	53M	114G	45.1	64.5	48.9	**28.4**	48.2	60.2

TABLE 6.6: Ablation study on the keypoint number M within each promising region. The sampling ratio r of prior detection predictions is set to 20% by default. Results are obtained on MS COCO val 2017.

M	#Params	FLOPs	AP	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L
1	53M	101G	43.9	64.3	47.5	25.1	46.9	60.8
2	53M	102G	45.0	64.7	48.9	26.0	48.3	60.4
4	53M	104G	45.3	**65.0**	48.7	**27.3**	48.1	60.9
8	53M	108G	**45.5**	**65.0**	**49.3**	**27.3**	48.3	**61.6**
16	53M	117G	45.3	64.7	49.0	26.6	**48.5**	61.5

6.6 Conclusion

This chapter presents *Iterative Multi-scale Feature Aggregation (IMFA)*, which defines the first generic paradigm for efficient use of multi-scale features in Transformer-based object detectors. IMFA exploits multi-scale features only from the most promising and informative locations and significantly improves detection accuracy on multiple object detectors at a marginal cost. The proposed IMFA bridges the gap in applying the newly proposed Transformer-based object detectors to leverage multi-scale features to deliver satisfactory detection accuracy on edge devices with limited computational capacity.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis presents four novel techniques for accurate and robust object detection in 2D images. In each of the research works, we have focused on various challenges in specific object detection setups and proposed corresponding solutions to address each difficulty or constraint, where contributions have been made to improve object detection performance greatly as well as to relieve various constraints (*e.g.*, a very limited number of annotated training samples, lack of visual clues for detection, limited resources for model training, *etc.*).

Specifically, Chapter 3 presents *Context-Aware Detection Network (CAD-Net)* to tackle the unique challenges in object detection for aerial remote sensing images. Unlike images captured by ground-level sensors, objects in remote sensing images often lack visual clues such as image contrast and texture details, are densely distributed and arbitrarily oriented, and suffer from severe noises due to various interference. To adapt to the constrained setups, the proposed CAD-Net extracts scene-level and object-level contextual information that is highly correlated to objects of interest to provide extra guidance for detecting objects within remote sensing images. In addition, a spatial-and-scale-aware attention module is also designed to guide the network to focus on scale-adaptive features and emphasizes the degraded texture details. These designs effectively compensate for the information loss within remote sensing images, leading to competitive performance under such constrained scenarios.

Chapter 4 studies the slow training convergence issue of a novel Transformer-based object detection framework (*i.e.*, DEtection TRansformer, DETR) [1]. DETR discards a series of hand-designed components such as anchors and non-maximum impression (NMS) and achieves fully end-to-end object detection in a simple pipeline, achieving competitive detection performance. However, DETR suffers from critically severe slow training convergence conventional ConvNet-based detectors like Faster R-CNN [21], which limits its broad applicability. In Chapter 4, we analyze that the root of DETR's slow convergence is largely attributed to the difficulty in matching object queries to relevant regions due to the unaligned semantics between object queries and encoded image features, and propose *Semantic-Aligned Matching DETR (SAM-DETR)* to accelerate DETR's convergence. The core of SAM-DETR is the semantic-aligned matching mechanism that projects object queries and encoded image features into the same feature embedding space, in which each object query can be easily matched to relevant regions with similar semantics. In addition, the semantic-aligned matching mechanism can be extended to fuse multi-scale features that are inherently unaligned semantics in a coarse-to-fine manner, resulting in *SAM-DETR++* with even faster convergence and higher detection accuracy. The proposed SAM-DETR and SAM-DETR++ alleviate the constraint of training Transformer-based object detectors with limited computational resources, reduce the carbon footprint, and pave the way for more comprehensive research and applications of Transformer-based object detectors.

Unlike most setups where abundant annotated training samples are available, Chapter 5 studies few-shot object detection – a special and challenging setup where only a few annotated samples are available for training, which is of great practical significance as abundant samples for training are often not available due to sample rarity, privacy concerns, as well as expensive labeling costs. Chapter 5 presents *Meta-DETR* for few-shot object detection, which adopts meta-learning to generalize base-class knowledge to novel classes. Meta-DETR bypasses the region proposal quality gap between base and novel classes, thus achieving superior performance than conventional R-CNN-based few-shot object detectors. In addition, Meta-DETR performs correlational meta-learning on a set of support classes at one go, thus effectively leveraging the inter-class correlation for better generalization, achieving state-of-the-art few-shot detection performance.

Furthermore, Chapter 6 studies the efficient use of multi-scale features in the

newly proposed Transformer-based object detectors. Multi-scale features have been proven highly effective for object detection, especially for small objects. Most ConvNet-based object detectors adopt Feature Pyramid Network (FPN) as the go-to component for exploiting multi-scale features. However, for the recently proposed Transformer-based object detectors, directly incorporating multi-scale features with FPN leads to prohibitive computational overhead due to the high complexity of the attention mechanism for processing high-resolution features. Chapter 6 presents *Iterative Multi-scale Feature Aggregation (IMFA)* to bridge this gap, which defines the first generic paradigm for efficient use of multi-scale features in Transformer-based object detectors. IMFA exploits multi-scale features only from the most promising and informative locations and significantly improves detection accuracy on multiple object detectors at a marginal cost. The proposed IMFA enables the newly proposed Transformer-based object detectors to leverage multi-scale features to deliver satisfactory detection performance even on edge devices with limited computational capacity.

7.2 Future Work

As a long-standing, fundamental, and challenging task in computer vision, object detection is of high research significance and is still facing many unsolved challenges.

Adaptation to Various Constrained Scenarios. As discussed in this thesis, when facing various constrained scenarios, generic object detection approaches inevitably suffer from performance drops. Typical challenging cases include occluded object detection, object detection with severe domain gap, few-shot object detection, dense object detection, long-tail object detection, *etc.* Designing appropriate algorithms accordingly is important for accurate and robust object detection under these constrained scenarios.

Self-Supervised Representation Learning. A good representation is the key to object detection. Recently, many studies [153, 154] have shown that self-supervised pretraining can produce powerful representations for various downstream tasks, even outperforming its supervised counterparts. Using self-supervised pertaining for object detection can reduce the amount of annotated data required

for training and enable the leverage of massive unlabeled data. Besides, the recent work DETReg [143] has shown promising results in self-supervised pertaining for object detection. We believe combining self-supervised representation learning will be highly beneficial for more accurate and robust object detection.

Multi-Modal Learning. Current object detectors [21, 26, 42, 43, 90] usually only take images as input, which requires extensive annotation labor in learning only a narrow set of visual concepts. Recently, a few research works have shown that multi-modal learning, especially combining vision with natural language processing (NLP), can achieve very competitive results. It is intuitive that NLP knowledge can supplement the learning of vision tasks. For example, SRR-FSD [65] shows that language semantics can effectively help to learn new visual concepts in the context of few-shot learning. Besides, CLIP [155] has demonstrated the strong power of the integration between computer vision and NLP. Therefore, we believe multi-modal learning combining knowledge from NLP has great potential to achieve more advanced object detection.

The recently proposed Transformer-based object detectors [1, 41–43, 138] are very suitable for multi-modal learning, because *(i)* similar to NLP models, they use Transformer as the building block; *(ii)* the Transformer attention mechanism is very suitable for aggregating multi-modal information. Therefore, we also highlight the importance of such newly proposed Transformer-based object detectors for their advantage for multi-modal learning.

Unifying Object Detection with Various Downstream Tasks. In practical setups, object detection often serves as a preliminary step in a pipeline and is crucial to many downstream tasks. However, cascading multiple stages is usually not optimal and may scale up the initial errors. An intuitive solution is to unify the task of object detection with other downstream tasks, making the system fully end-to-end. This avoids the accumulation of errors in the initial stage of object detection and thus can improve the robustness of those detection-involved vision systems.

In conclusion, object detection is a fundamental task in computer vision, which is of great significance. Although object detection has achieved unprecedented progress with the development of deep neural networks, it is still far from a solved problem. There is still room for further performance improvement of generic object detection.

Besides, object detection still requires non-trivial adaptation when facing various constrained scenarios. Furthermore, there is massive potential for object detection with self-supervised representation learning and multi-modal learning. We will continue to work on this topic in the future.

List of Author's Patent and Publications

Patent

- **Gongjie Zhang**, Kaiwen Cui, Shijian Lu, and Tzu-Yi Hung. "Defect Sample Synthesis Method, Training Method of Defect Inspection Network, Computer-Readable Medium, Computing System, and Image Inspection Apparatus." China Patent No.: CN114693595A (Issued Jul. 2022); Singapore Patent Application No.: 10202114457P (Filed Dec. 2021).

Journal Articles

- **Gongjie Zhang**, Shijian Lu, and Wei Zhang. "CAD-Net: A Context-Aware Detection Network for Objects in Remote Sensing Imagery." *IEEE Transactions on Geoscience and Remote Sensing (**T-GRS**)*, vol. 57, no. 12, pp. 10015-10024, 2019. (DOI: 10.1109/TGRS.2019.2930982)

- **Gongjie Zhang**, Zhipeng Luo, Kaiwen Cui, Shijian Lu, and Eric P. Xing. "Meta-DETR: Image-Level Few-Shot Detection with Inter-Class Correlation Exploitation." *IEEE Transactions on Pattern Analysis and Machine Intelligence (**T-PAMI**)*, 2022. (DOI: 10.1109/TPAMI.2022.3195735)

Conference Proceedings

- Rongliang Wu, **Gongjie Zhang**, Shijian Lu, and Tao Chen. "Cascade EF-GAN: Progressive Facial Expression Editing with Local Focuses." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (Oral Presentation)*, 2020.

- **Gongjie Zhang**, Kaiwen Cui, Rongliang Wu, Shijian Lu, and Yonghong Tian. "PNPDet: Efficient few-shot detection without forgetting via plug-and-play sub-networks." In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021.

- **Gongjie Zhang**, Kaiwen Cui, Tzu-Yi Hung, and Shijian Lu. "Defect-GAN: High-fidelity defect synthesis for automated defect inspection." In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021.

- Zhipeng Luo, Zhongang Cai, Changqing Zhou, **Gongjie Zhang**, Haiyu Zhao, Shuai Yi, Shijian Lu, Hongsheng Li, Shanghang Zhang, and Ziwei Liu. "Unsupervised Domain Adaptive 3D Detection with Multi-Level Consistency." In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

- Kaiwen Cui, Jiaxing Huang, Zhipeng Luo, **Gongjie Zhang**, Fangneng Zhan, and Shijian Lu. "GenCo: Generative Co-training for Generative Adversarial Networks with Limited Data." In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2022.

- **Gongjie Zhang**, Zhipeng Luo, Yingchen Yu, Kaiwen Cui, and Shijian Lu. "Accelerating DETR Convergence via Semantic-Aligned Matching." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.

Bibliography

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. xxi, xxii, 4, 15, 16, 41, 44, 45, 46, 49, 53, 54, 55, 56, 57, 58, 60, 61, 62, 69, 80, 93, 97, 101, 102, 104, 107, 109, 110, 111, 112, 113, 118, 120

[2] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3212–3232, 2019. 1

[3] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019. 9, 10

[4] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *International Journal of Computer Vision*, 128:261–318, 2020. 1, 9, 41, 97

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 2, 9, 11, 16, 18, 22, 33, 66, 81

[6] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. 2, 3, 17, 18, 22, 42, 44, 55, 66, 81, 107, 109

[7] R. Lienhart and J. Maydt. An extended set of Haar-like features for rapid object detection. In *Proceedings of the International Conference on Image Processing (ICIP)*, 2002. 10

[8] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 11

[9] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 10

[10] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001. 10, 12

[11] Pedro F Felzenszwalb, Ross B Girshick, and David McAllester. Cascade object detection with deformable part models. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 11, 13

[12] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 11, 13

[13] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. Ensemble of exemplar-SVMs for object detection and beyond. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011. 11

[14] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9): 1627–1645, 2010. 11, 13

[15] Ross Girshick, Pedro Felzenszwalb, and David McAllester. Object detection with grammar models. *Advances in Neural Information Processing Systems*, 2011. 11, 12, 13

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012. 12, 13, 35

[17] 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 (CVPR)*, 2014. 13, 22, 30, 69

[18] 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(2):154–171, 2013. 13

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, 2015. 13

[20] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. 13, 22, 25, 30, 69

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017. 13, 14, 16, 22, 23, 24, 30, 31, 34, 35, 37, 38, 41, 42, 45, 52, 55, 57, 58, 60, 61, 66, 69, 70, 74, 83, 84, 97, 100, 112, 113, 118, 120

[22] 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 (CVPR)*, 2017. 14, 19, 25, 31, 35, 38, 42, 52, 55, 58, 60, 61, 97, 101, 102, 108, 113

[23] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 48, 66, 76, 93, 101, 102

[24] 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 (CVPR)*, 2018. 58, 61, 100

[25] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 19, 97

[26] Gongjie Zhang, Shijian Lu, and Wei Zhang. CAD-Net: A context-aware detection network for objects in remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 57(12):10015–10024, 2019. 21, 69, 97, 101, 120

[27] Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xiaodan Liang, and Liang Lin. Meta R-CNN: Towards general solver for instance-level low-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 14, 19, 70, 73, 74, 80, 81, 83, 84, 93, 94, 100

[28] 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 (CVPR)*, 2016. 14, 15, 22, 30, 41, 42, 69, 97

[29] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 15, 22, 30, 34, 41, 42, 45, 69, 97

[30] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 14, 15

[31] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 15, 22, 30, 34, 45, 69, 97, 101

[32] 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 (ICCV)*, 2017. 15, 42, 55, 80, 101

[33] Hei Law and Jia Deng. CornerNet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, 2018. 15

[34] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 15, 45, 69, 97

[35] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 15, 16, 41, 42, 45, 55, 58, 66, 69, 97, 101

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, L. Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. 15, 41, 44, 49, 53, 63, 64, 75, 78, 97, 101, 104

[37] Mingxing Tan, Ruoming Pang, and Quoc V Le. EfficientDet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 16, 52, 66, 69, 97, 101, 108

[38] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 16, 19, 42, 43, 45, 46, 49, 53, 55, 56, 57, 58, 60, 61, 75, 80, 82, 83, 84, 97, 98, 101, 102, 104, 111, 113

[39] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional DETR for fast training convergence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 43, 44, 49, 53, 57, 58, 60, 61, 69, 104, 107, 109, 110, 111, 112, 113

[40] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor DETR: Query design for Transformer-based detector. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022. 42, 55, 58, 60, 61, 104, 109, 111, 112, 113

[41] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. DAB-DETR: Dynamic anchor boxes are better queries for DETR. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. 45, 46, 49, 55, 56, 58, 61, 69, 97, 102, 104, 107, 109, 110, 111, 112, 113, 114, 120

[42] Gongjie Zhang, Zhipeng Luo, Yingchen Yu, Kaiwen Cui, and Shijian Lu. Accelerating DETR convergence via semantic-aligned matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 41, 53, 69, 99, 104, 111, 113, 120

[43] Gongjie Zhang, Zhipeng Luo, Kaiwen Cui, Shijian Lu, and Eric P. Xing. Meta-DETR: Image-level few-shot detection with inter-class correlation exploitation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. doi: 10.1109/TPAMI.2022.3195735. 43, 45, 53, 69, 101, 120

[44] Changqing Zhou, Zhipeng Luo, Yueru Luo, Tianrui Liu, Liang Pan, Zhongang Cai, Haiyu Zhao, and Shijian Lu. PTTR: Relational 3D point cloud object tracking with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 16

[45] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 17

[46] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 17, 33, 55, 110

[47] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, 128(7):1956–1981, 2020. 17

[48] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 17

[49] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. ICDAR2013 robust reading competition. In *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR)*, 2013. 17

[50] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. ICDAR2015 competition on robust reading. In *Proceedings of the 13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015.

[51] Baoguang Shi, Cong Yao, Minghui Liao, Mingkun Yang, Pei Xu, Linyan Cui, Serge Belongie, Shijian Lu, and Xiang Bai. ICDAR2017 competition on reading Chinese text in the wild (RCTW-17). In *Proceedings of the 14th*

International Conference on Document Analysis and Recognition (ICDAR), 2017.

[52] Chee-Kheng Chng, Yuliang Liu, Yipeng Sun, Chun Chet Ng, Canjie Luo, Zihan Ni, ChuanMing Fang, Shuaitao Zhang, Junyu Han, Errui Ding, Jingtuo Liu, Dimosthenis Karatzas, Chee Seng Chan, and Lianwen Jin. ICDAR2019 robust reading challenge on arbitrary-shaped text (RRC-ArT). In *Proceedings of the 15th International Conference on Document Analysis and Recognition (ICDAR)*, 2019. 17

[53] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive Faster R-CNN for object detection in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 19

[54] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 100

[55] Yangtao Zheng, Di Huang, Songtao Liu, and Yunhong Wang. Cross-domain object detection through coarse-to-fine feature adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR*, pages 13766–13775, 2020. 19

[56] Xue Yang, Hao Sun, Kun Fu, Jirui Yang, Xian Sun, Menglong Yan, and Zhi Guo. Automatic ship detection in remote sensing images from Google Earth of complex scenes based on multiscale rotation dense feature pyramid networks. *Remote Sensing*, 10(1):132, 2018. 19, 34, 101

[57] Xue Yang, Jirui Yang, Junchi Yan, Yue Zhang, Tengfei Zhang, Zhi Guo, Xian Sun, and Kun Fu. SCRDet: Towards more robust detection for small, cluttered and rotated objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8232–8241, 2019.

[58] Xue Yang and Junchi Yan. Arbitrary-oriented object detection with circular smooth label. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 677–694, 2020.

[59] Xue Yang, Junchi Yan, Qi Ming, Wentao Wang, Xiaopeng Zhang, and Qi Tian. Rethinking rotated object detection with Gaussian Wasserstein distance loss. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 11830–11841, 2021. 19

[60] Xue Yang, Hao Sun, Xian Sun, Menglong Yan, Zhi Guo, and Kun Fu. Position detection and direction prediction for arbitrary-oriented ships via multiscale rotation region convolutional neural network. *arXiv preprint arXiv:1806.04828*, 2018. 19, 34

[61] Hao Chen, Yali Wang, Guoyou Wang, and Yu Qiao. LSTD: A low-shot transfer detector for object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 19, 73, 83, 100

[62] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 70, 73, 74, 81, 83, 84

[63] Xin Wang, Thomas Huang, Joseph Gonzalez, Trevor Darrell, and Fisher Yu. Frustratingly simple few-shot object detection. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020. 73, 81, 82, 83, 84, 100

[64] Yang Xiao and Renaud Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 70, 73, 74, 81, 82, 83, 84, 85, 86, 87, 100

[65] Chenchen Zhu, Fangyi Chen, Uzair Ahmed, Zhiqiang Shen, and Marios Savvides. Semantic relation reasoning for shot-stable few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 83, 84, 120

[66] Weilin Zhang and Yu-Xiong Wang. Hallucination improves few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 19, 83, 84, 100

[67] Tao Wang, Li Yuan, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. PnP-DETR: Towards efficient visual analysis with Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 19, 101, 102

[68] Byungseok Roh, JaeWoong Shin, Wuhyun Shin, and Saehoon Kim. Sparse DETR: Efficient end-to-end object detection with learnable sparsity. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. 53, 61, 98, 101, 102, 113

[69] Chenhongyi Yang, Zehao Huang, and Naiyan Wang. QueryDet: Cascaded sparse query for accelerating high-resolution small object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 19, 102, 107

[70] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: Object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems*, 2016. 22, 34, 69, 70, 97

[71] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. DOTA: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), 2018. 22, 32, 34, 35

[72] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015. 26

[73] Fan Yang, Ke Yan, Shijian Lu, Huizhu Jia, Xiaodong Xie, and Wen Gao. Attention driven person re-identification. *Pattern Recognition*, 86:143–155, 2019. 26

[74] Gong Cheng, Junwei Han, Peicheng Zhou, and Lei Guo. Multi-class geospatial object detection and geographic image classification based on collection of part detectors. *ISPRS Journal of Photogrammetry and Remote Sensing*, 98:119–132, 2014. 32, 34, 35

[75] Gong Cheng and Junwei Han. A survey on object detection in optical remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 117: 11–28, 2016. 32, 34, 35

[76] 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 (CVPR)*, 2016. 33, 34, 38, 55, 57, 62, 82, 110, 112

[77] Seyed Majid Azimi, Eleonora Vig, Reza Bahmanyar, Marco Körner, and Peter Reinartz. Towards multi-class object detection in unconstrained remote sensing imagery. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2018. 34

[78] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 34

[79] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015. 34

[80] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 34

[81] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 35

[82] Gong Cheng, Peicheng Zhou, and Junwei Han. Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12): 7405–7415, 2016. 35

[83] Ke Li, Gong Cheng, Shuhui Bu, and Xiong You. Rotation-insensitive and context-augmented object detection in remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 56(4):2337–2348, 2018. 35

[84] Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of DETR with spatially modulated co-attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 42, 43, 44, 45, 46, 49, 53, 54, 55, 57, 58, 60, 61, 65, 69, 70, 97, 98, 101, 102, 104, 107, 113

[85] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. DN-DETR: Accelerate DETR training by introducing query denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 43, 44, 55, 61, 65, 101

[86] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional Siamese networks for object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 43

[87] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with Siamese region proposal network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[88] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. SiamRPN++: Evolution of Siamese visual tracking with very deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[89] Paul Voigtlaender, Jonathon Luiten, Philip HS Torr, and Bastian Leibe. Siam R-CNN: Visual tracking by re-detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 100

[90] Ning Wang, Wengang Zhou, Jie Wang, and Houqiang Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 120

[91] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 43

[92] Dahjung Chung, Khalid Tahboub, and Edward J Delp. A two stream Siamese convolutional neural network for person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 43

[93] Meng Zheng, Srikrishna Karanam, Ziyan Wu, and Richard J Radke. Re-identification with consistent attentive Siamese networks. In *CVPR*, 2019.

[94] Lin Wu, Yang Wang, Junbin Gao, and Xue Li. Where-and-when to look: Deep Siamese attention networks for video-based person re-identification. *IEEE Transactions on Multimedia*, 21(6):1412–1424, 2018.

[95] Chen Shen, Zhongming Jin, Yiru Zhao, Zhihang Fu, Rongxin Jiang, Yaowu Chen, and Xian-Sheng Hua. Deep Siamese network with multi-level similarity perception for person re-identification. In *Proceedings of the 25th ACM international conference on Multimedia (ACM MM)*, 2017.

[96] Yantao Shen, Tong Xiao, Hongsheng Li, Shuai Yi, and Xiaogang Wang. Learning deep neural networks for vehicle Re-ID with visual-spatio-temporal path proposals. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 43

[97] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, 2015. 43

[98] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017.

[99] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[100] Ting-I Hsieh, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. One-shot object detection with co-attention and co-excitation. In *Advances in Neural Information Processing Systems*, 2019. 43, 100

[101] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krähenbühl. Bottom-up object detection by grouping extreme and center points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 44, 49, 99

[102] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. RepPoints: Point set representation for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[103] Yihong Chen, Zheng Zhang, Yue Cao, Liwei Wang, Stephen Lin, and Han Hu. RepPoints v2: Verification meets regression for object detection. In *Advances in Neural Information Processing Systems*, 2020. 49, 99

[104] Qijie Zhao, Tao Sheng, Yongtao Wang, Zhi Tang, Ying Chen, Ling Cai, and Haibin Ling. M2Det: A single-shot object detector based on multi-level feature pyramid network. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2019. 52, 69, 70, 97, 101, 108

[105] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research (JMLR)*, 15(1):1929–1958, 2014. 53, 64

[106] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 55, 82, 110

[107] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 55, 82, 110

[108] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, and Ping Luo. Sparse R-CNN: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 58, 61, 66, 69, 70, 108, 113

[109] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. UP-DETR: Unsupervised pre-training for object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 60, 61, 101

[110] Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris M. Kitani. Rethinking Transformer-based set prediction for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 61, 112, 113

[111] Barbara Landau, Linda B Smith, and Susan S Jones. The importance of shape in early lexical learning. *Cognitive development*, 3(3):299–321, 1988. 69

[112] Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual Review of Biomedical Engineering*, 19(1):221–248, 2017. 70

[113] Suiyi Ling, Andreas Pastor, Jing Li, Zhaohui Che, Junle Wang, Jieun Kim, and Patrick Le Callet. Few-shot pill recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[114] Gongjie Zhang, Kaiwen Cui, Tzu-Yi Hung, and Shijian Lu. Defect-GAN: High-fidelity defect synthesis for automated defect inspection. In *Proceedings*

of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2021. 70

[115] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-RPN and multi-relation detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 70, 73, 74, 84, 100

[116] Hanzhe Hu, Shuai Bai, Aoxue Li, Jinshi Cui, and Liwei Wang. Dense relation distillation with context-aware aggregation for few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 70, 73, 83, 84, 100

[117] Weilin Zhang, Yu-Xiong Wang, and David A Forsyth. Cooperating RPN's improve few-shot object detection. *arXiv preprint arXiv:2011.10142*, 2020. 70, 74

[118] Jiaxi Wu, Songtao Liu, Di Huang, and Yunhong Wang. Multi-scale positive sample refinement for few-shot object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 73, 83, 84, 100

[119] Bo Sun, Banghuai Li, Shengcai Cai, Ye Yuan, and Chi Zhang. FSCE: Few-shot object detection via contrastive proposal encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 73, 83, 84

[120] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Meta-learning to detect rare objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 73, 81, 83, 84, 100

[121] Juan-Manuel Perez-Rua, Xiatian Zhu, Timothy M Hospedales, and Tao Xiang. Incremental few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 73

[122] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 80

[123] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 80

[124] Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Sharathchandra Pankanti, Rogerio Feris, Abhishek Kumar, Raja Giries, and Alex M Bronstein. RepMet: Representative-based metric learning for classification and one-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 83, 100

[125] Zhibo Fan, Yuchen Ma, Zeming Li, and Jian Sun. Generalized few-shot object detection without forgetting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 83

[126] Bohao Li, Boyu Yang, Chang Liu, Feng Liu, Rongrong Ji, and Qixiang Ye. Beyond max-margin: Class margin equilibrium for few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 83, 84, 100

[127] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. 87

[128] Gangming Zhao, Weifeng Ge, and Yizhou Yu. GraphFPN: Graph feature pyramid network for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 97, 101, 108

[129] Seung-Wook Kim, Hyong-Keun Kook, Jee-Young Sun, Mun-Cheon Kang, and Sung-Jea Ko. Parallel feature pyramid network for object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[130] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 101

[131] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[132] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. NAS-FPN: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 97, 101, 108

[133] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra R-CNN: Towards balanced learning for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 100

[134] Lachlan Tychsen-Smith and Lars Petersson. Improving object localization with fitness NMS and bounded IoU loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[135] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[136] Bharat Singh, Mahyar Najibi, and Larry S Davis. SNIPER: Efficient multi-scale training. In *Advances in Neural Information Processing Systems*, 2018.

[137] Minghui Liao, Pengyuan Lyu, Minghang He, Cong Yao, Wenhao Wu, and Xiang Bai. Mask TextSpotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(2):532–548, 2021. 100

[138] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. MDETR–modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 101, 120

[139] Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu. You only look at one sequence: Rethinking transformer in vision through object detection. In *Advances in Neural Information Processing Systems*, 2021. 113

[140] Xipeng Cao, Peng Yuan, Bailan Feng, and Kun Niu. CF-DETR: Coarse-to-fine transformers for end-to-end object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022. 101

[141] Ishan Misra, Rohit Girdhar, and Armand Joulin. An End-to-End Transformer Model for 3D Object Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[142] Akshita Gupta, Sanath Narayan, KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. OW-DETR: Open-world detection transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[143] 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 (CVPR)*, 2022. 120

[144] Hwanjun Song, Deqing Sun, Sanghyuk Chun, Varun Jampani, Dongyoon Han, Byeongho Heo, Wonjae Kim, and Ming-Hsuan Yang. ViDT: An efficient and effective fully transformer-based object detector. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. 113

[145] Wen Wang, Yang Cao, Jing Zhang, and Dacheng Tao. FP-DETR: Detection transformer advanced by fully pre-training. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. 101

[146] Mikhail Figurnov, Aizhan Ibraimova, Dmitry P Vetrov, and Pushmeet Kohli. PerforatedCNNs: Acceleration through elimination of redundant convolutions. In *Advances in Neural Information Processing Systems*, 2016. 102

[147] Michael Figurnov, Maxwell D Collins, Yukun Zhu, Li Zhang, Jonathan Huang, Dmitry Vetrov, and Ruslan Salakhutdinov. Spatially adaptive computation time for residual networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[148] Mengye Ren, Andrei Pokrovsky, Bin Yang, and Raquel Urtasun. SBNet: Sparse blocks network for fast inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[149] Thomas Verelst and Tinne Tuytelaars. Dynamic convolutions: Exploiting spatial sparsity for faster inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 102

[150] Mahyar Najibi, Bharat Singh, and Larry S Davis. AutoFocus: Efficient multi-scale inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 102, 107

[151] Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient DETR: improving end-to-end object detector with dense prior. *arXiv preprint arXiv:2104.01318*, 2021. 112, 113

[152] 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 (ICCV)*, 2021. 112

[153] 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 (CVPR)*, 2020. 119

[154] 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 (CVPR)*, 2022. 119

[155] 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 *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021. 120