



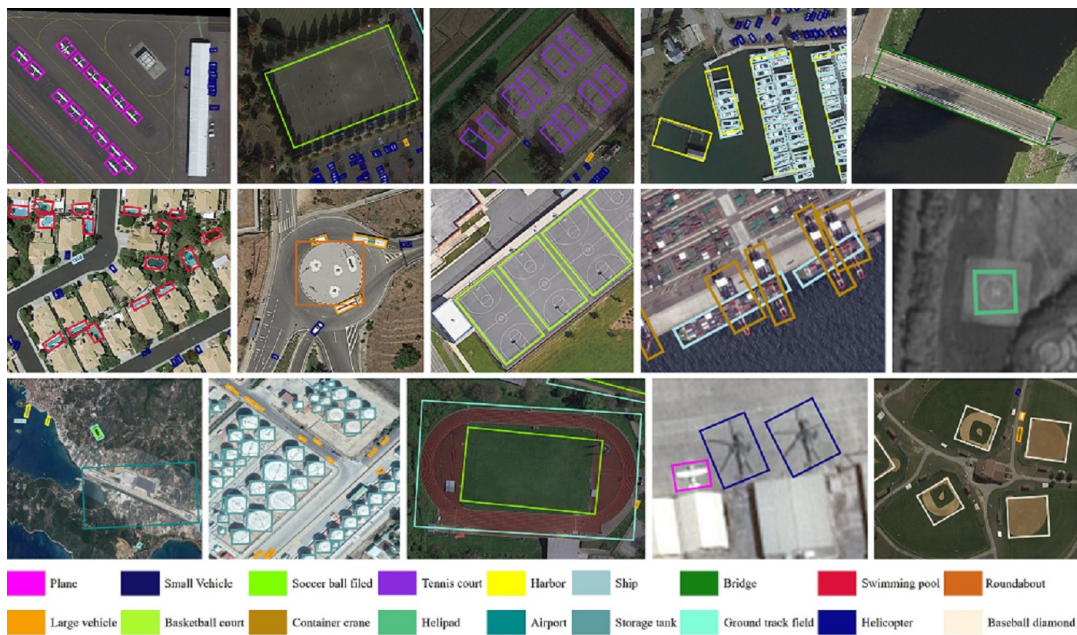
NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
MSc DATA SCIENCE & MACHINE LEARNING

Transfer Learning for Deep Object Detectors in Remote Sensing Imaging Datasets

DIPLOMA THESIS

of

PAVLOS KALFANTIS



Supervisor: Konstantinos Karantzas
Professor, NTUA

Athens, July 2023



NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
MSc DATA SCIENCE & MACHINE LEARNING

Transfer Learning for Deep Object Detectors in Remote Sensing Imaging Datasets

DIPLOMA THESIS

of

PAVLOS KALFANTIS

Supervisor: Konstantinos Karantzas
Professor, NTUA

Approved by the examination committee on 12th July, 2023.

(Signature)

(Signature)

(Signature)

.....
Konstantinos Karantzas
Professor, NTUA

.....
Giorgos Stamou
Professor, NTUA

.....
Athanasios Voulodimos
Assistant Professor, NTUA

Athens, July 2023



Copyright © - All rights reserved.

Pavlos Kalfantis, 2023.

The copying, storage and distribution of this diploma thesis, exall or part of it, is prohibited for commercial purposes. Reprinting, storage and distribution for non - profit, educational or of a research nature is allowed, provided that the source is indicated and that this message is retained.

The content of this thesis does not necessarily reflect the views of the Department, the Supervisor, or the committee that approved it.

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism.

(Signature)

.....
Pavlos Kalfantis

12th July, 2023

Abstract

Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. Object detection specifically in aerial and satellite images presents unique challenges compared to object detection in natural images, as aerial and satellite images often suffer from object size, scale and resolution variations, complex backgrounds and imbalanced datasets. Additionally, the large size of these images poses computational challenges for efficient and accurate object detection algorithms. Finally, the publicly available remote sensing imaging datasets are limited and creating and annotating new ones is a time and resource consuming endeavor.

Transfer learning is a technique in machine learning in which knowledge learned from a task is re-used in order to boost performance on a related task. Reusing/transferring information from previously learned tasks to new tasks has the potential to significantly improve learning efficiency, as it allows the models to converge faster and potentially achieve better performance, even with a limited amount of data for the new task. In this thesis, the technique of transfer learning of pre-trained object detectors on a large dataset to new datasets with or without further training is investigated. The rationale behind this work concerns the study of the performance of the trained object detectors when evaluated on similar categories of new datasets, in order to identify the challenges and strengths of this approach as a solution to the challenges of object detection in satellite and aerial images.

The current thesis is divided into three main parts. The first part of the thesis formulates the problem and challenges of object detection in remote sensing imagery, as well as it explains how transfer learning can be used to tackle these challenges. In addition, it investigates the current state-of-the-art object detection algorithms, where a variety of models are presented and five models are selected to be used in experiments throughout this thesis.

In the next part of the thesis, the method that is followed during this project is formulated. The transfer learning approach that is followed is analyzed and the datasets that are being used in the experiments are presented. In addition, the metrics that are used as a basis for the evaluation of our experiments are explained.

In the final part of the thesis, the results of the experiments are presented and analyzed. The results of training the object detectors in the baseline dataset, as well as the results of transferring the trained object detectors to two new datasets with similar objects are presented both quantitatively and qualitatively. Finally, conclusions regarding the method that was used and the results that were obtained are drawn and summarized.

Περίληψη

Η ανίχνευση αντικειμένων είναι μια τεχνολογία που σχετίζεται με την όραση υπολογιστών και την επεξεργασία εικόνας που ασχολείται με την ανίχνευση αντικειμένων μιας συγκεκριμένης κατηγορίας (όπως άνθρωποι, κτίρια ή αυτοκίνητα) σε ψηφιακές εικόνες και βίντεο. Η ανίχνευση αντικειμένων ειδικά σε εναέριες και δορυφορικές εικόνες παρουσιάζει μοναδικές προκλήσεις σε σύγκριση με την ανίχνευση αντικειμένων σε φυσικές εικόνες, καθώς οι εναέριες και δορυφορικές εικόνες συχνά έχουν μεγάλες διακυμάνσεις μεγέθους αντικειμένων, κλίμακας και ανάλυσης, πολύπλοκα υπόβαθρα ενώ τα σύνολα δεδομένων είναι μη ισορροπημένα. Επιπλέον, το μεγάλο μέγεθος αυτών των εικόνων θέτει υπολογιστικές προκλήσεις για αποτελεσματικούς και ακριβείς αλγόριθμους ανίχνευσης αντικειμένων. Τέλος, τα διαθέσιμα σύνολα δεδομένων είναι περιορισμένα και η δημιουργία νέων συνόλων δεδομένων καταναλώνει χρόνο και πόρους.

Η μεταφορά μάθησης είναι μια τεχνική στη μηχανική μάθηση στην οποία η γνώση που αποκτάται σέ ένα πρόβλημα επαναχρησιμοποιείται προκειμένου να ενισχυθεί η απόδοση σε ένα σχετικό πρόβλημα. Η μεταφορά πληροφοριών σε νέα προβλήματα έχει τη δυνατότητα να βελτιώσει σημαντικά την αποτελεσματικότητα της εκμάθησης, καθώς επιτρέπει στα μοντέλα να συγκλίνουν ταχύτερα και ενδεχομένως να επιτύχουν καλύτερη απόδοση, ακόμη και με περιορισμένο αριθμό νέων δεδομένων. Σε αυτή τη διπλωματική εργασία, διερευνάται η τεχνική μεταφοράς μάθησης ανιχνευτών αντικειμένων, προεκπαιδευμένων σε ένα μεγάλο σύνολο δεδομένων, σε νέα σύνολα δεδομένων με ή χωρίς περαιτέρω εκπαίδευση. Ο στόχος σε αυτήν την εργασία περιλαμβάνει τη μελέτη της απόδοσης των εκπαιδευμένων ανιχνευτών αντικειμένων όταν αξιολογούνται σε παρόμοιες κατηγορίες νέων συνόλων δεδομένων, προκειμένου να εντοπιστούν οι προκλήσεις και τα πλεονεκτήματα της ανίχνευσης αντικειμένων σε δορυφορικές και εναέριες εικόνες .

Η παρούσα διπλωματική εργασία χωρίζεται σε τρία μέρη. Το πρώτο μέρος της εργασίας διατυπώνει το πρόβλημα και τις προκλήσεις της ανίχνευσης αντικειμένων στις εικόνες τηλεπισκόπησης, καθώς και εξηγεί πώς μπορεί να χρησιμοποιηθεί η μεταφορά μάθησης για την αντιμετώπιση αυτών των προκλήσεων. Επιπλέον, διερευνά τους σύγχρονους αλγόριθμους ανίχνευσης αντικειμένων, όπου παρουσιάζονται διαφορετικά σύγχρονα μοντέλα και επιλέγονται πέντε μοντέλα που θα χρησιμοποιηθούν στα πειράματα.

Στο επόμενο μέρος της διπλωματικής εργασίας, διατυπώνεται η μέθοδος που ακολουθείται κατά τη διάρκεια αυτής της εργασίας. Αναλύεται η μέθοδος μεταφοράς μάθησης που ακολουθείται και παρουσιάζονται τα σύνολα δεδομένων που χρησιμοποιούνται στα πειράματα. Επιπλέον, εξηγούνται οι μετρικές που χρησιμοποιούνται ως βάση για την αξιολόγηση των πειραμάτων.

Στο τελευταίο μέρος της διπλωματικής εργασίας παρουσιάζονται και αναλύονται τα απο-

τελέσματα των πειραμάτων. Παρουσιάζονται τόσο ποσοτικά όσο και ποιοτικά τα αποτελέσματα της εκπαίδευσης των ανιχνευτών αντικειμένων στο βασικό σύνολο δεδομένων, καθώς και τα αποτελέσματα της μεταφοράς των εκπαιδευμένων ανιχνευτών αντικειμένων σε δύο νέα σύνολα δεδομένων με παρόμοια αντικείμενα. Τέλος, εξάγονται συμπεράσματα σχετικά με τη μέθοδο που χρησιμοποιήθηκε και τα αποτελέσματα που προέκυψαν.

to my parents

Acknowledgements

I would like to express my sincere gratitude to my supervisor prof. Konstantinos Karantzalos, for giving me the opportunity to work on this interesting field and for providing me with his feedback whenever I needed it. I'm also extremely grateful to PhD candidates Athena Psalta and Vasilis Tsironis, for their constant support since day one, and their willingness to help and provide directions. The completion of this thesis would not be possible without them. Finally, I would like to thank my family for their love and encouragement and for always believing in me.

Athens, July 2023

Pavlos Kalfantis

Contents

Abstract	1
Περίληψη	3
Acknowledgements	7
1 Introduction	15
1.1 Object Detection in Remote Sensing Imaging	15
1.2 Transfer Learning	16
1.3 Thesis Outline and Structure	17
2 Background	19
2.1 History of Object Detection	19
2.2 Two-Stage Detectors	20
2.2.1 Faster-RCNN	20
2.3 One-Stage Detectors	23
2.3.1 RetinaNet	23
2.3.2 YOLO	24
2.3.3 YOLOX	25
2.4 Transformer Based Object Detection	26
2.4.1 DETR	26
2.4.2 Deformable DETR	27
2.4.3 ConvNeXt Architecture	27
3 Proposed Method	31
3.1 Transfer Learning	31
3.2 Evaluation Metrics	32
3.2.1 Precision and Recall	32
3.2.2 Intersection over Union	32
3.2.3 Average Precision and mean Average Precision	33
3.3 Datasets	34
3.3.1 DOTA	34
3.3.2 HRRSD	35
3.3.3 ITCVD	36
3.4 Training and Evaluation Procedure	37

4 Results and Evaluation	39
4.1 Results on DOTAv2	39
4.2 Results on HRRSD	40
4.2.1 Quantitative Results	40
4.2.2 Qualitative Results	41
4.3 Results on ITCVD	43
4.3.1 Quantitative Results	43
4.3.2 Qualitative Results	44
5 Conclusions	61
Bibliography	65
List of Abbreviations	67

List of Figures

2.1	Object Detection Milestones from survey [1]	20
2.2	Differences of One Stage and Two Stage Detectors	20
2.3	Faster R-CNN Architecture	21
2.4	Example of 9 possible Anchors in an input image	22
2.5	Region Proposal Network Architecture	22
2.6	Fast R-CNN Architecture	23
2.7	RetinaNet Architecture	24
2.8	YOLO Architecture	25
2.9	Difference between YOLOv3 head and YOLOX decoupled head.	26
2.10	DETR High-Level Architecture	26
2.11	Deformable DETR High-Level Architecture	28
2.12	Deformable Attention used by the Deformable DETR Algorithm	29
2.13	Differences between a standard ResNet, a SWIN Transformer and ConvNeXt block	30
3.1	Intersection over Union definition	33
3.2	True Positive and False Positive definition in relation with IoU	33
3.3	Examples of DOTA annotations for each category	36
4.1	Training Loss in DOTA v2	46
4.2	Transfer Learning mAP50 curve for different percentages of training set	47
4.3	Transfer Learning mAP50:95 curve for different percentages of training set	47
4.4	Test Image from HRRSD with ground truth annotations - Case 1	48
4.5	Test Image from HRRSD with ground truth annotations - Case 2	48
4.6	Test Image from HRRSD with ground truth annotations - Case 3	49
4.7	Test Image from HRRSD with ground truth annotations - Case 4	49
4.8	Test Image from HRRSD with ground truth annotations - Case 5	50
4.9	Test Image from HRRSD with ground truth annotations - Case 6	50
4.10	Object Detections in HRRSD - Case 1	51
4.11	Object Detections in HRRSD - Case 2	52
4.12	Object Detections in HRRSD - Case 3	53
4.13	Object Detections in HRRSD - Case 4	54
4.14	Object Detections in HRRSD - Case 5	55
4.15	Object Detections in HRRSD - Case 6	56
4.16	Test Image from ITCVD - Case 1	57

4.17	Test Image from ITCVD - Case 2	57
4.18	Zero-shot Detections in ITCVD patch - Case 1	58
4.19	Zero-shot Detections in ITCVD patch - Case 2	59
4.20	Zero-shot Detections in ITCVD full image with Faster R-CNN - Case 1	60
4.21	Zero-shot Detections in ITCVD full image with Faster R-CNN - Case 2	60

List of Tables

3.1	Number of Objects in DOTAv2	35
3.2	Number of Objects in HRRSD	37
3.3	Number of Images in the different HRRSD datasets	37
4.1	Results of DOTAv2	40
4.2	mAP50 Results on HRRSD	42
4.3	mAP50:95 Results on HRRSD	43
4.4	Classwise AP50:95 on HRRSD - Zero-shot	43
4.5	Classwise AP50:95 on HRRSD - 0.5% of training set	44
4.6	Classwise AP50:95 on HRRSD - 1% of training set	44
4.7	Classwise AP50:95 on HRRSD - 5% of training set	45
4.8	Classwise AP50:95 on HRRSD - 10% of training set	45
4.9	Classwise AP50:95 on HRRSD - 20% of training set	45
4.10	AP50 Results on ITCVD	45
4.11	AP50:95 Results on ITCVD	46

Introduction

1.1 Object Detection in Remote Sensing Imaging

Remote sensing concerns the acquisition of information about an object or phenomenon without making physical contact with the object, in contrast to in situ or on-site observation. Today, remote sensing technologies enable the observation of the earth's surface using aerial images, with very high resolution. In recent years, deep learning and specifically computer vision advancements have transformed the way satellite and aerial images are analyzed and interpreted. A key example of a computer vision related task that is widely used in order to analyze and interpret remote sensing images is object detection. Object detection refers to the localization of objects of interest along with the prediction of the category of the localized object in images. Object detection is a key component in many computer vision applications such as automated driving, video surveillance, and object tracking. It is one of the most researched areas of computer vision and in recent years, several neural network-based approaches have been developed that are able to detect objects in images with very high precision. However, object detection when applied in aerial and satellite images still faces numerous challenges compared to object detection in standard images. Some of the key important challenges are:

- **Large Image Sizes:** Aerial and satellite images are often extremely large for current GPU memory capacity to analyze.
- **Scale and Resolution Variations:** Satellite images can have significant scale variations, as objects of interest may vary in size depending on their distance from the satellite. Additionally, different satellites or sensors can capture images at varying resolutions, leading to challenges in detecting objects accurately across different scales. Finally aerial images can be taken in different angles, from nadir view to oblique view.
- **Object Size Variations and Small Object Detection:** Object instances vary dramatically in scale. This is not only because of the spatial resolutions of sensors, but also due to the size variations inside the same object category. Many aerial images also contain small objects which are heavily overwhelmed by complex surrounding scenes.

- **Crowded Objects:** Objects such as vehicles in parking lots and planes in airports are often densely arranged, leading to feature coupling between classes due to overlapping. Similarly to large image sizes, objects appearing in large multitudes cannot be processed by current GPUs due to insufficient memory capabilities.
- **Class Imbalance:** Remote sensing imaging datasets tend to exhibit class imbalance, meaning that certain object classes may be underrepresented compared to others. This often leads to biased model training and affect the performance of object detection algorithms, particularly for rare or less frequent classes.
- **Complex Backgrounds and Clutter:** Aerial and satellite images frequently exhibit complex backgrounds due to the presence of diverse land cover, natural features, and human infrastructure. Distinguishing objects from cluttered backgrounds can be difficult, especially when objects share similar visual characteristics with the surrounding environment.
- **Limited Labeled Data:** Annotated datasets for object detection in remote sensing images are limited compared to other domains. Collecting and labeling large-scale satellite imaging datasets is a resource-intensive and time-consuming process. The scarcity of labeled data poses challenges for training accurate and robust object detection models that can be applied in different tasks effectively.

1.2 Transfer Learning

The machine learning technique of transfer learning can help tackle and overcome some of the challenges of object detection in remote sensing images that were presented in the previous section. Transfer learning is a technique in machine learning in which knowledge learned from one task is re-used in order to boost performance on a related task. So, if a pre-trained model that has been trained on a large-scale dataset for object detection is available, this model can be re-used in a different task.

The general steps of a transfer learning pipeline that can be applied in remote sensing imaging start with selecting an object detector model. State-of-the-art models like Faster R-CNN, RetinaNet, YOLO or DETR that will be presented in chapter 2 can serve as good starting points for this task. The next step is to acquire a dataset of satellite and aerial images with annotations in the form of bounding boxes around the objects classes of interest. After that, data preprocessing steps like radiometric and atmospheric corrections, data augmentation and image resizing and normalization are standard practice in this type of images. Then the model is trained and finetuned on this preprocessed annotated dataset until the results of the detections are deemed satisfactory.

The final step is to apply the technique of transfer learning to new data and evaluate the robustness of the pre-trained model. There are several transfer learning strategies that can be used:

- **Full Transfer Learning:** Fine-tune the entire network, including the backbone layers, using the new dataset. This strategy requires a sufficient amount of labeled data.

- **Partial Transfer Learning:** Freeze some of the initial layers (backbone layers) of the pre-trained model while only training the new task-specific layers on the target dataset. This approach is suitable when the target dataset is limited.
- **Few-Shot Learning:** Fine-tune the network with a very small number of labeled samples, even as low as one or a few examples per class. This approach is useful when only a limited amount of annotated data is available.
- **Zero-Shot Learning:** Evaluate the pre-trained model without further training on the new dataset and assess its performance.

Finally, in order to evaluate the performance of the transfer learning-based object detector on a separate validation or test set, appropriate evaluation metrics such as mean Average Precision (mAP), Intersection over Union (IoU), or Precision-Recall curves can be used. The results can be compared with baseline models or other approaches to measure the improvement achieved through transfer learning.

By applying transfer learning in object detection for satellite and aerial images, the model can benefit from the rich representations learned from large-scale datasets, enhancing the precision of object detections. This approach can mitigate the challenges listed and explained in the previous section, leading to more accurate and robust object detection capabilities. Chapter 3 presents and explains the method of transfer learning that is used in this thesis and the datasets that on which our method is evaluated.

1.3 Thesis Outline and Structure

The current thesis is organized as follows:

- **Chapter 2 - Background:** This chapter presents the history of object detection and explains several state-of-the-art algorithms for object detection. Finally, five models are selected for the experiments that were conducted in later parts of this thesis.
- **Chapter 3 - Proposed Method:** This chapter formulates the method that was used in the experiments of this thesis, regarding transfer learning from pre-trained object detectors to different datasets. In addition, the evaluation metrics that were used in the experiments are formulated and explained. Finally, the three datasets that were used in the experiments as well as the training and evaluation procedure are presented.
- **Chapter 4 - Results and Evaluation:** This chapter presents all experiments that were conducted related to Object Detection. First, the results of training the five proposed models on the baseline dataset are presented. Then, these pre-trained models are used in new datasets with or without further training and the results are presented both qualitatively and quantitatively.
- **Chapter 5 - Conclusions:** This chapter includes a quick summary of the work that was presented as well as some key conclusions.

Background

2.1 History of Object Detection

Object detection is an important computer vision task that deals with detecting instances of visual objects of a certain class (such as humans, animals, or cars) in digital images. The goal of object detection is to develop computational models and techniques that provide one of the most basic pieces of knowledge needed by computer vision applications: What objects are where? The two most significant metrics for object detection are accuracy (including classification accuracy and localization accuracy) and speed. Object detection serves as a basis for many other computer vision tasks, such as instance segmentation image captioning, object tracking etc. In recent years, the rapid development of deep learning techniques has greatly promoted the progress of object detection, leading to remarkable breakthroughs and propelling it to a research hotspot with unprecedented attention. Object detection has now been widely used in many real-world applications, such as autonomous driving, robot vision, video surveillance, multiple object tracking etc.

The progress of object detection has generally gone through two historical periods: “Traditional object detection period (before 2014)” and “Deep learning based detection period (after 2014)”, as shown in figure 2.1. The traditional object detection methods were based on handcrafted features. Due to the lack of effective image representation at that time, people had to design sophisticated feature representations and a variety of speed-up skills. Some of the most important detectors of this period were the Viola Jones Detectors [2], the HOG Detector [3] and the Deformable Part-based Model [4].

With the rapid development of the deep Convolutional Neural Networks for computer vision tasks, many algorithms for object detection (deep object detectors) started appearing in research. In 2014 R. Girshick et al. proposed the Region-Based Convolutional Networks (R-CNN) [5], which was the first deep object detector that improved the detection capabilities of previous algorithm dramatically. This led to models Fast R-CNN [6] and eventually Faster R-CNN [7] that is considered state-of-the-art even today. In the era of deep learning, object detection can be grouped in two categories, “two-stage detectors” (like Faster R-CNN) and “one-stage detectors” (like YOLO [8], RetinaNet [9] and DETR [10]), where the former frames the detection as a “coarse-to-fine” process while the latter frames it as to “complete in one step”. More specifically, the two stage process first generates region proposals and then classifies each proposal into different object classes. On the

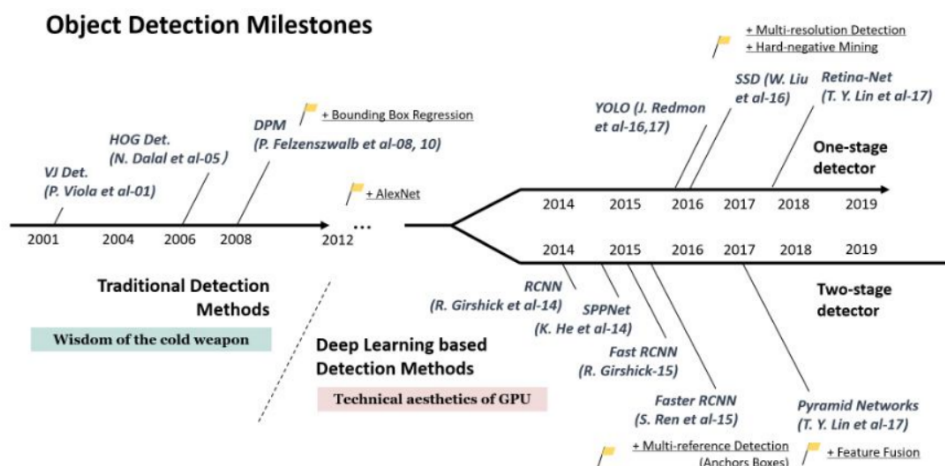


Figure 2.1. Object Detection Milestones from survey [1]

other hand, one-stage detection considers object detection as a simple joint regression and classification problem, adopting a unified framework to acquire final object classes and locations in one step. The differences of the two approaches are depicted in figure 2.2. In the next sections of this chapter, several state-of-the-art models are presented and analyzed. Then, five of these models will eventually be trained and evaluated with the proposed method of transfer learning in this thesis.

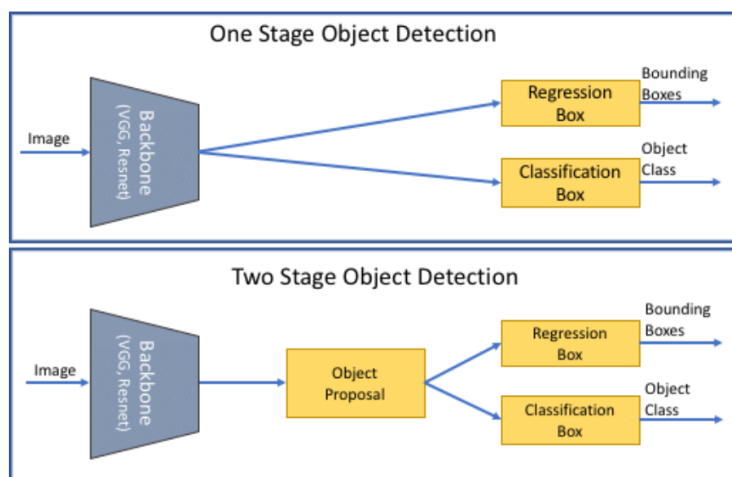


Figure 2.2. Differences of One Stage and Two Stage Detectors

2.2 Two-Stage Detectors

2.2.1 Faster-RCNN

Faster R-CNN (Region-based Convolutional Neural Network) is a popular two-stage object detection framework that combines the benefits of deep convolutional neural net-

works (CNNs) and region proposal methods. It was introduced by Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun in 2015 [7] as an extension of the original R-CNN [6]. Figure 2.3 depicts the architecture of the Faster R-CNN algorithm.

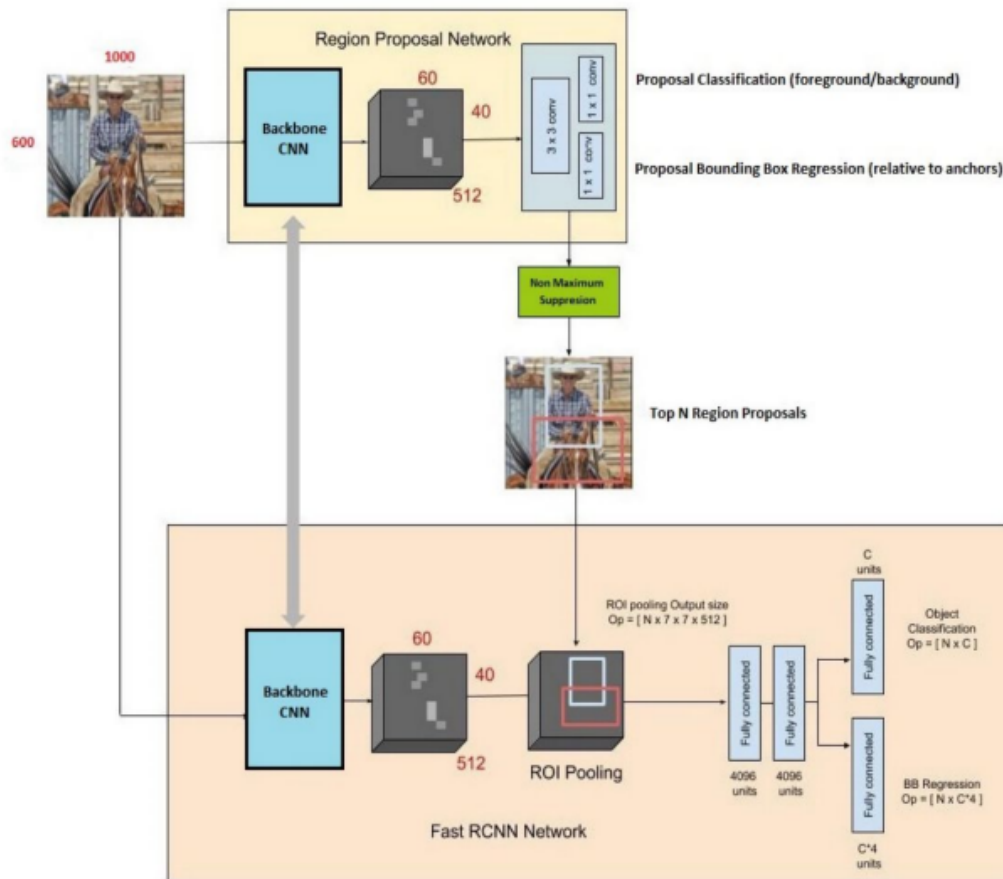


Figure 2.3. *Faster R-CNN Architecture*

The first stage of the algorithm concerns the creation of region proposals (Region Proposal Network - RPN), which are potential bounding boxes that contain objects in the image. It achieves this by sliding a small network, typically a CNN, over the convolutional feature map produced by a shared CNN backbone network. Popular CNN backbones that are typically used for the RPN are ResNet [11] or VGG [12]. The predecessor algorithm Fast R-CNN uses the Selective Search algorithm for Object Proposals instead of the RPN, which is a time consuming process. In contrast, the RPN predicts objectness scores and bounding box regression offsets for a set of anchor boxes at different positions and scales. Anchors are placed on the initial image to depict possible objects and various sizes and aspect ratios, with an example shown on figure 2.4. Since anchors usually overlap, proposals also end up overlapping. Thus, these predicted proposals are then ranked and filtered using non-maximum suppression (NMS) to obtain the most likely object regions. The overall Region Proposal Network architecture is depicted in figure 2.5.

The second stage of the algorithm is identical with the Fast R-CNN detector. The

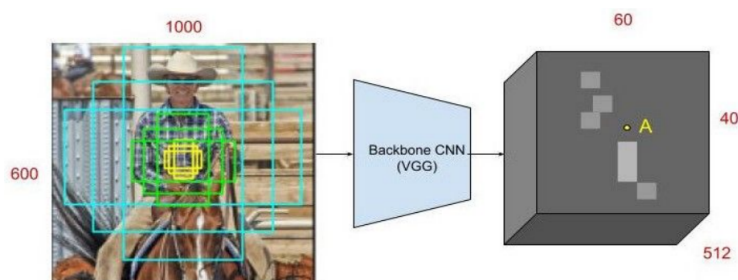


Figure 2.4. Example of 9 possible Anchors in an input image

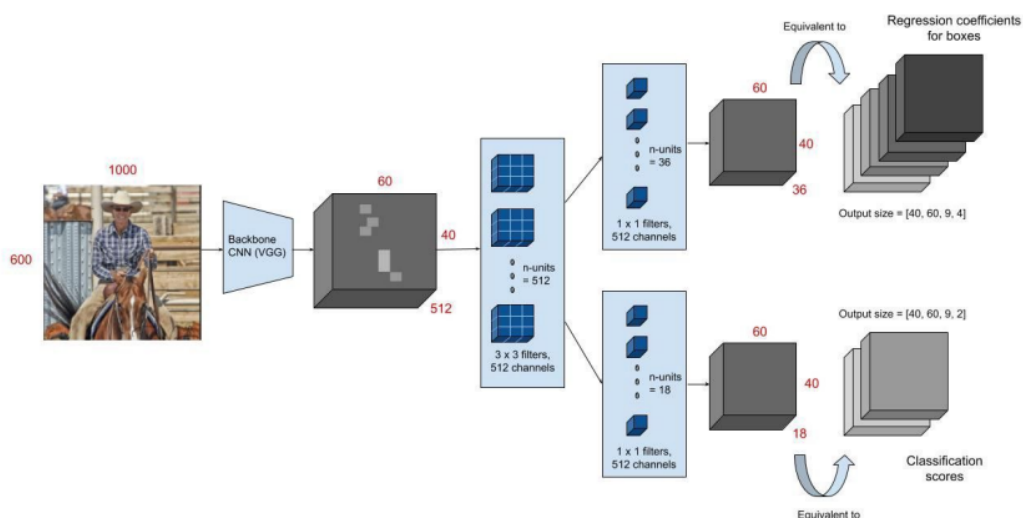


Figure 2.5. Region Proposal Network Architecture

architecture of the Fast R-CNN is depicted in figure 2.6. The Fast R-CNN architecture includes a backbone CNN and a Region of Interest (ROI) layer which isolates the region of the feature map that corresponds to the Object Proposal. These two connected layers are split into 2 branches for the classification of the object and the regression of the bounding box. An important aspect is that the classification and regression parts of the Fast R-CNN algorithm are different from the RPN classification and regression.

The two-stage detector is trained in 4 steps. First, The RPN network is independently trained on the Region Proposal problem. A pre-trained network is used as a backbone and is then fine-tuned. Second, the Fast R-CNN part is independently trained using the proposed regions of the RPN on the Detection problem. The next step is to initialize the RPN network using the weights of the Fast R-CNN, so that the network is fine-tuned on the Region Proposal task. The weights on the common layers of RPN and Fast R-CNN are kept frozen. The final step involves training again the layers of the detection part of Fast R-CNN on the final trained RPN.

Faster R-CNN is a very powerful algorithm in the task of object detection. For this thesis, a Faster R-CNN model using the ResNet50 CNN as backbone is selected as the

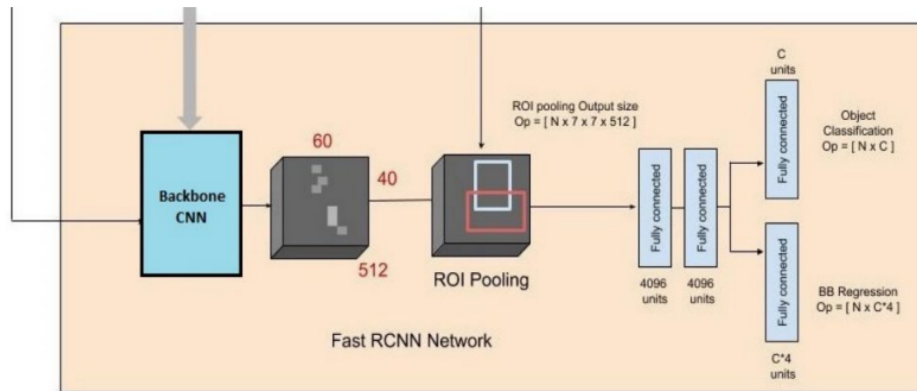


Figure 2.6. *Fast R-CNN Architecture*

first model for the experiments.

2.3 One-Stage Detectors

2.3.1 RetinaNet

RetinaNet [9] is a powerful Single Shot Detector that is comprised of three distinct parts. A pre-trained backbone encoder CNN, a Feature Pyramid Network (FPN) and Convolutional heads to detect the objects built. The high-level architecture of RetinaNet is depicted in figure 2.7. The main innovation of RetinaNet lies in its feature pyramid network (FPN) backbone and the use of a novel focal loss function, as explained below.

The Feature Pyramid Network is the first key innovation and the basis upon which RetinaNet is built. It starts with a backbone network (usually ResNet or VGG similar to Faster R-CNN), in order to extract features from the input image. The FPN is then applied on top to generate a pyramid of feature maps with different spatial resolutions. The FPN combines high-resolution, semantically strong features with low-resolution, semantically weak features through a top-down pathway and lateral connections. This process creates a rich set of multi-scale features that are critical for detecting objects of various sizes.

The other key innovation of RetinaNet is the use of focal loss. One key challenge in object detection is handling the extreme class imbalance between the vast number of background regions and the limited number of foreground object regions. To address this, RetinaNet introduces a novel loss function called the focal loss. The focal loss assigns higher weights to hard and misclassified examples while down-weights easy examples. By focusing on challenging samples, the model effectively addresses the class imbalance issue and improves the accuracy of object detection.

In the final stage, RetinaNet includes two parallel sub-networks, the classification sub-network (head) and the regression sub-network (head). The classification head predicts the probability of each anchor box containing an object belonging to various predefined classes. It applies several convolutional layers followed by a classification layer. The regression head estimates the precise bounding box coordinates (i.e., offsets) for each

anchor box, adjusting them to tightly fit the objects in the image.

During training, RetinaNet optimizes both the classification and regression heads using the focal loss. The model is trained end-to-end, where the backbone network, FPN, and the two sub-networks are jointly optimized.

RetinaNet offers several advantages. It achieves impressive accuracy by effectively handling object detection at multiple scales with the help of the FPN. The focal loss improves the model's ability to handle class imbalance, leading to better detection performance. Furthermore, RetinaNet operates efficiently by sharing computation across all spatial locations of the image, making it also suitable for real-time applications.

Overall, RetinaNet has become a popular choice in the field of object detection due to its accuracy, efficiency, and ability to handle multi-scale detection effectively. It has been widely adopted in various domains, including autonomous driving, robotics, and surveillance systems. For this thesis, a RetinaNet model using the ResNet50 CNN as backbone is selected as the first model for the experiments.

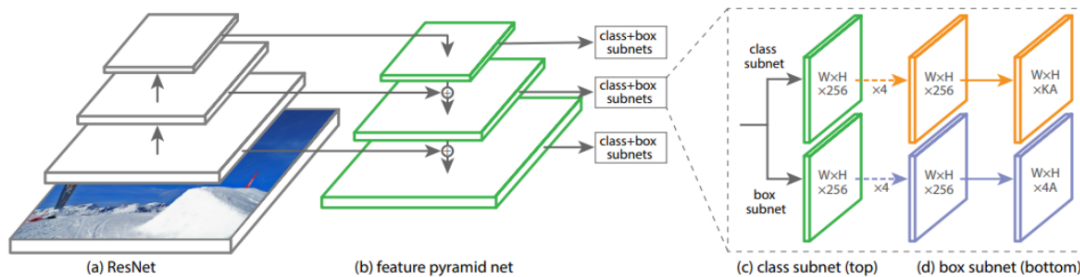


Figure 2.7. RetinaNet Architecture

2.3.2 YOLO

You Only Look Once (YOLO) was a novel approach to object detection that was introduced in 2015 [8]. It frames object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. It uses a single neural network to predict bounding boxes and associated class probabilities directly from the full images in one evaluation.

YOLO starts by dividing the input image into a grid of N cells, with each one being responsible to detect and localize the object that it contains. These cells predict the coordinates of the bounding box (Bounding Box Regression) and the class probability and confidence score that an object is inside them (Classification). Finally it uses Non-Maximum Suppression (NMS) to eliminate duplicate and overlapping detections. NMS selects the most confident detection for each object and suppresses overlapping detections based on a threshold, resulting in a final set of non-overlapping, high-confidence detections. The architecture of the YOLO algorithm is depicted in figure 2.8.

Regarding the loss function used in YOLO, a complex function is used. It contains 2 terms for bounding box regression, 2 terms for cell classification. Finally there is a common term for classification and localization of the prediction. The loss function used

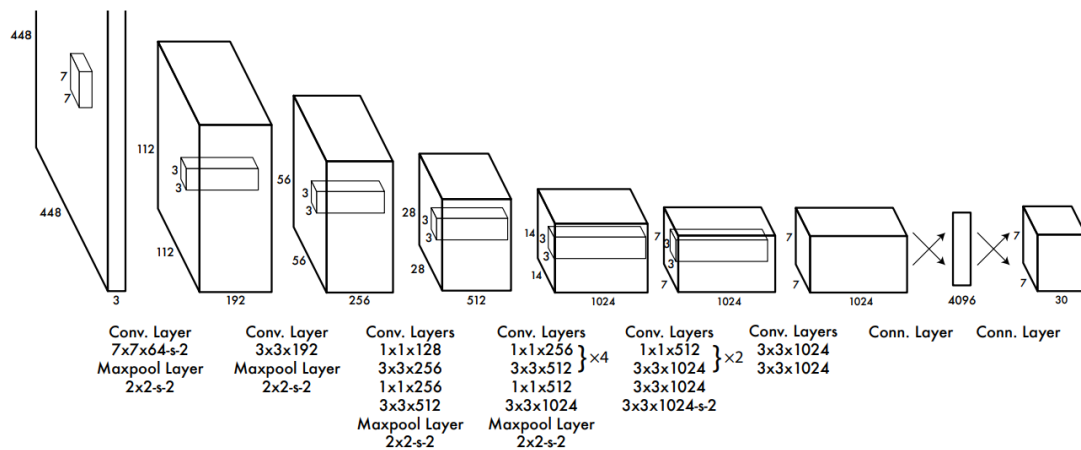


Figure 2.8. YOLO Architecture

in YOLO combines the localization loss (based on the bounding box coordinates) and the classification loss (based on the class probabilities). The confidence score is used to filter out low-confidence detections during post-processing.

YOLO offers several advantages, including very high real-time performance, simplicity, and the ability to detect objects accurately in different domains. It has been widely adopted in various applications, such as autonomous driving, video surveillance, and object recognition. However, YOLO may face challenges in detecting small objects and objects with significant aspect ratio variations due to the fixed grid cell size and anchor boxes. Finally, the loss function treats the same way errors in small and large bounding boxes.

2.3.3 YOLOX

A series of algorithms have been developed since the inception of YOLO, like YOLOv2 [13], YOLOv3 [14], YOLOv4 and YOLOv5. Each one of them built on top of the original algorithm to tackle its limitations and problems. The YOLO series always pursuit the optimal speed and accuracy trade-off for real-time applications. However, all of these detectors didn't follow the major advances in object detection, i.e. anchor-free detectors, advanced label assignment strategies and end-to-end (without NMS) detectors. The latest iteration of the YOLO series that included these advances was the YOLOX model [15].

The key difference in YOLOX is that the original algorithm was switched to an anchor free manner, while also adding a decoupled head and a novel label assignment strategy (SimOTA). The difference in the architecture of YOLOX decoupled and the head of previous iterations of the YOLO algorithms is depicted in figure 2.9.

YOLOX was able to achieve a better trade-off between speed and accuracy in all model sizes. It surpassed the average precision achieved by the YOLOv3 algorithm while also being able to achieve faster real-time detections. As a result, the YOLOX algorithm was selected as the third model of the experiments of this thesis.

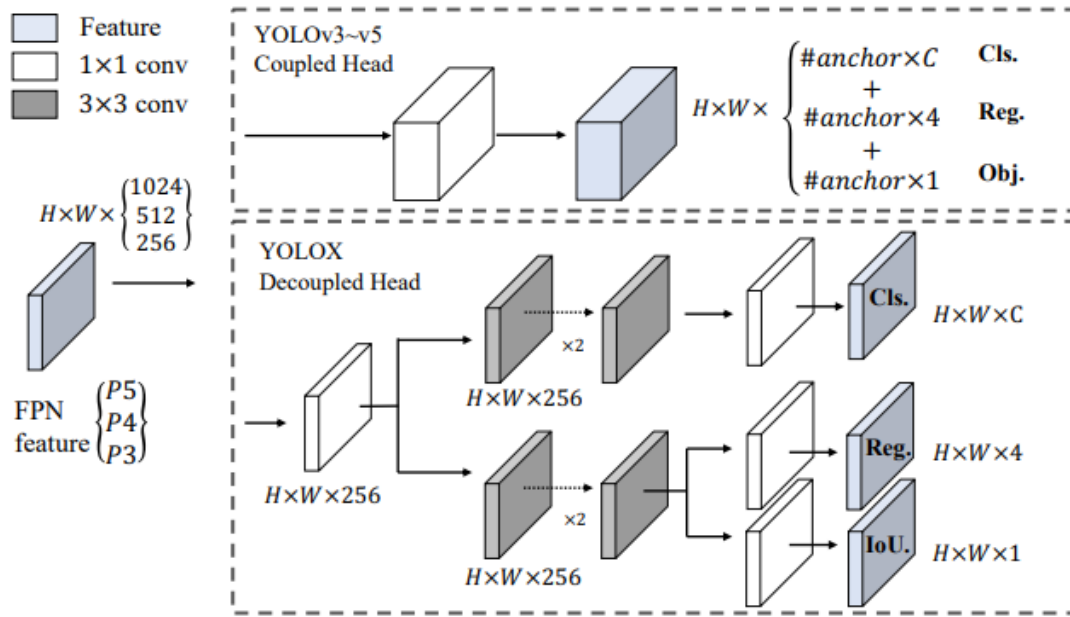


Figure 2.9. Difference between YOLOv3 head and YOLOX decoupled head.

2.4 Transformer Based Object Detection

2.4.1 DETR

The DETR Algorithm is based on the paper 'End-to-end Object Detection with Transformers' by META AI[10]. Unlike traditional object detection models that rely on region proposal methods and complex pipelines, DETR leverages the Transformer Architecture [16] to perform end-to-end object detection in a single pass. It represents a significant departure from the conventional two-stage approaches and has gained attention for its simplicity, efficiency, and impressive performance. The high level architecture of the DETR algorithm is depicted in figure 2.10.

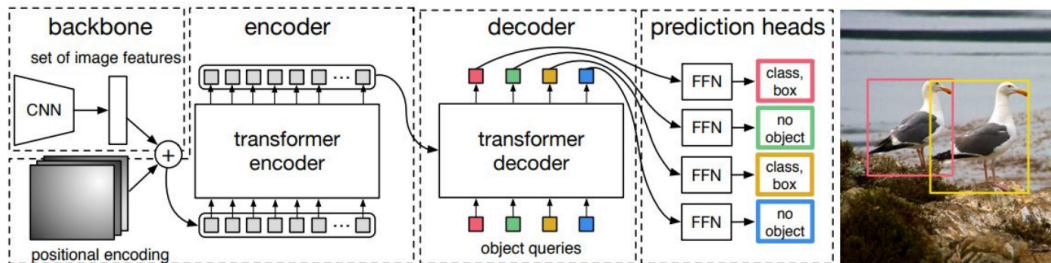


Figure 2.10. DETR High-Level Architecture

The problem of object detection in the DETR algorithm is modeled as an image-to-set problem. This eliminates the need for algorithms that include hand-crafted features like the non-maximum suppression. In addition, the anchors that are basic on the architecture of Faster R-CNN and RetinaNet to model the prior knowledge of the problem

are not used. It is based on a single Transformer Encoder-Decoder architecture and a set-based global loss function that results to unique predictions with the technique of bipartite matching.

The transformer encoder processes the input image by dividing it into a grid of equally-sized spatial regions or patches. Each patch is passed through a convolutional neural network (CNN) backbone to extract a feature vector. These features are then flattened and processed by a stack of transformer encoder layers. The encoder layers capture both local and global contextual information, enabling effective representation learning.

The transformer decoder takes the output of the encoder, which consists of a set of fixed-length feature vectors, and generates object queries and positional encodings. The object queries serve as the representation of the classes to be detected. The positional encodings capture the spatial information of each object query. The decoder layers attend to both the encoder outputs and the previously generated object queries to refine the object representations iteratively.

DETR offers several advantages over traditional object detection models. It eliminates the need for complex region proposal methods, such as selective search or anchor-based sampling, simplifying the detection pipeline. The end-to-end nature of DETR enables efficient training and inference. Furthermore, DETR demonstrates impressive performance, achieving competitive results on various benchmark datasets.

2.4.2 Deformable DETR

Deformable DETR is an extension of the original DETR framework that incorporates deformable convolutional networks (DCN) to improve the modeling of spatial relationships and handle object deformations more effectively. It was introduced by Xizhou Zhu et al. in 2021 [17]. The high level architecture of the Deformable DETR algorithm is depicted in figure 2.11.

The deformable convolutional networks are introduced by Deformable DETR as the backbone network in the encoder, in order to enhance the modelling of spatial relationships by allowing the convolutional filter to be spatially adaptive. In addition Deformable DETR introduces deformable attention mechanisms in the transformer decoder, in order to guide the attention computations, as depicted in figure 2.12.

The additions of deformable convolutional networks and attention mechanisms are the key improvements to the DETR algorithm. These additions enhance the model's ability to handle object deformations and improve the accuracy of object detection. Deformable DETR represents a significant advancement in the field of object detection, particularly for scenarios where objects undergo significant shape changes or deformations. For this thesis, a Deformable DETR model is selected as the fourth model for the experiments.

2.4.3 ConvNeXt Architecture

The introduction of Vision Transformers (ViTs) [18] in 2020 were a breakthrough in the field of computer vision. They quickly superseded Convolutional Neural Networks as the state-of-the-art image classification model. However, they faced difficulties when applied

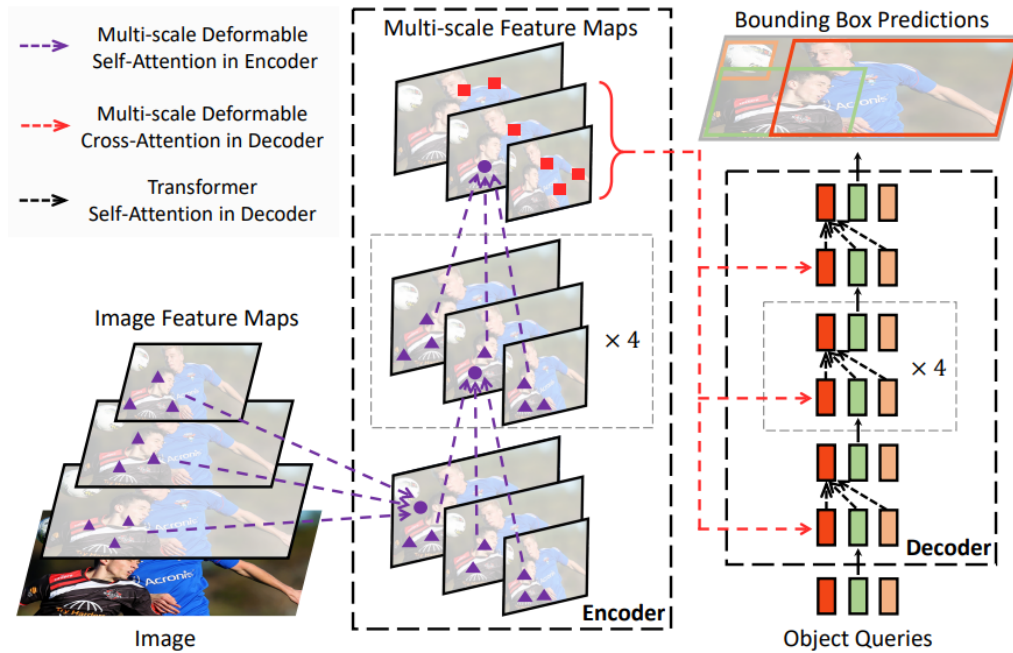


Figure 2.11. *Deformable DETR High-Level Architecture*

to different computer vision tasks, such as object detection and semantic segmentation. This led to the introduction of the hierarchical Transformers (e.g., Swin Transformers [19]) that reintroduced several Convolutional Net priors, making Transformers practically viable as a generic vision backbone and demonstrating remarkable performance on a wide variety of vision tasks beyond image classification. Swin Transformer’s success and rapid adoption also revealed one thing: the essence of convolution is not becoming irrelevant; rather, it remains much desired and has never faded.

However, the effectiveness of such hybrid approaches was still largely credited to the intrinsic superiority of Transformers, rather than the inherent inductive biases of convolutions. Then Liu et. al in their work [20] reexamined the design spaces and tested the limits of what a pure ConvNet can achieve. The result was to gradually "modernize" a standard ResNet toward the design of a vision Transformer, and discover several key components that contribute to the performance difference along the way. The outcome of this exploration was a family of pure ConvNet models dubbed ConvNeXt. Constructed entirely from standard ConvNet modules, ConvNeXts compete favorably with Transformers in terms of accuracy and scalability outperforming Swin Transformers on detection segmentation tasks, while maintaining the simplicity and efficiency of standard Convolutional Neural Networks. Figure 2.13 shows the difference of the standard block of a classic ConvNet (ResNet), a SWIN Transformer and the newly introduced ConvNeXt model.

This inspired us to explore a new model for the experiments done in this thesis, where the standard ResNet50 backbone network of the Faster R-CNN is replaced by a ConvNeXt based model. The resulting model is named Faster R-CNN ConvNeXt and serves as the fifth model in the experiments that will be presented and evaluated in the following

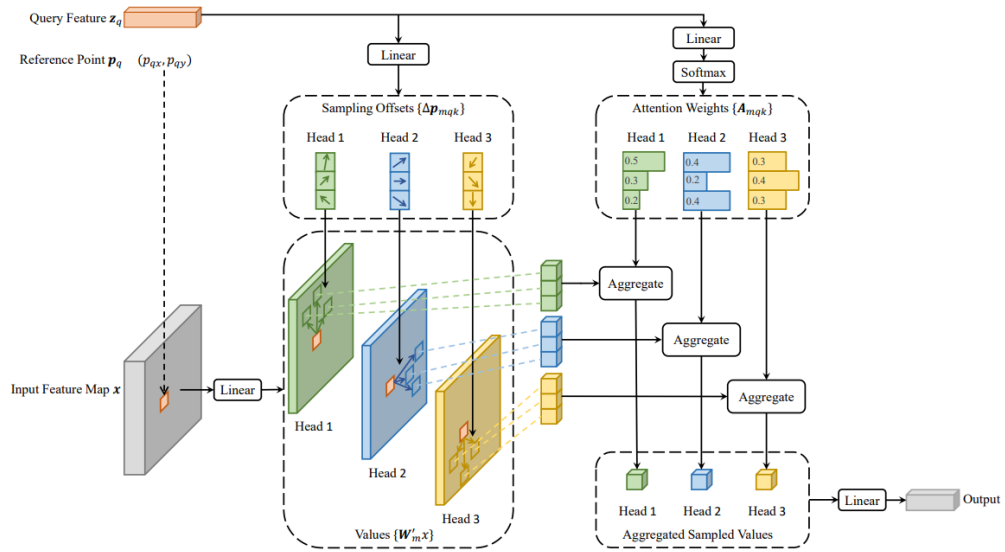


Figure 2.12. Deformable Attention used by the Deformable DETR Algorithm

chapters.

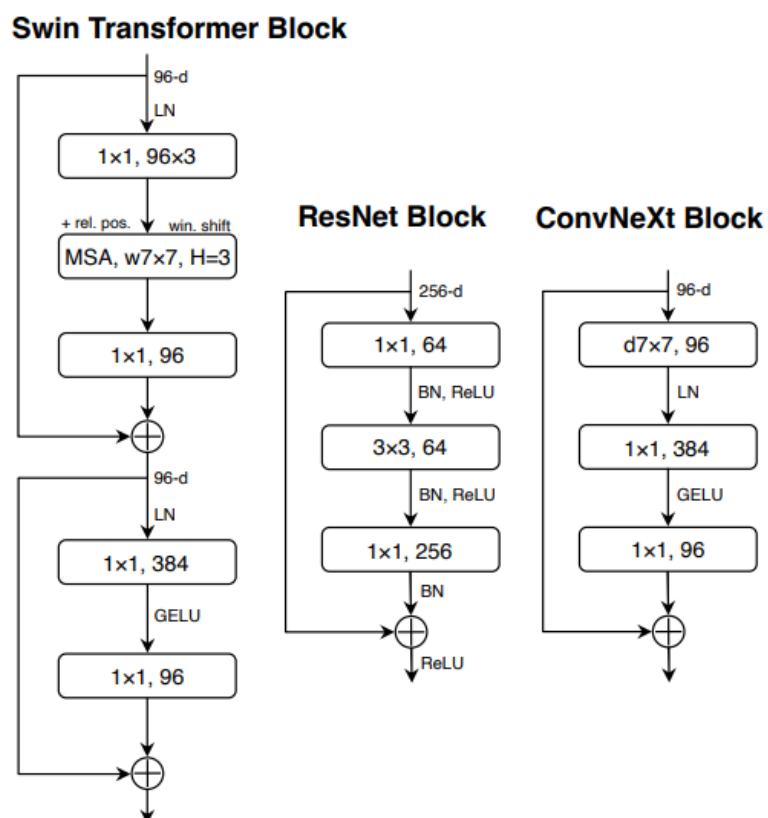


Figure 2.13. Differences between a standard ResNet, a SWIN Transformer and ConvNeXt block

Chapter **3**

Proposed Method

In this chapter, the proposed method for the experiments conducted during this thesis is presented. We present the method of Transfer Learning that is used, as well as the Evaluation Metrics for the experiments. We also present the three datasets that were used for the experiments and some information on the training and evaluation procedures.

3.1 Transfer Learning

The rationale behind the work done during this thesis is based primarily on the technique of transfer learning. Transfer learning is a widely used technique in machine learning in which knowledge learned from a task is re-used in order to boost performance on a related task. The idea is that reusing/transferring information from previously learned tasks to new tasks has the potential to significantly improve learning efficiency.

The task at hand was identifying objects in satellite and aerial images, which inherently contains many challenges, as explained in chapter 1. In summary, different datasets contain images with varying scales, object sizes and resolutions or are taken at a different angle. In addition, there are limited labeled datasets with aerial images and creating new ones is expensive and time consuming. To that end, the proposed approach is to train the models presented in chapter 2 in a very large, publicly available and well annotated dataset with many object categories (presented in section 3.3.1), in order to have models that are capable of identifying the objects that this dataset contains. Subsequently, these models can be evaluated on other datasets that contain similar object categories, like the ones presented in subsections 3.3.2 and 3.3.3 and evaluate them in the same categories. Going a step further, we evaluate the models on the new datasets with two different approaches. First, we evaluate them without further training (zero-shot learning) and secondly, we evaluate them while further training them on small subsets of the new dataset (few-shot learning). In the second approach, we experiment with different small subsets of the new dataset in order to identify the number of training images from the new dataset that can be proven sufficient for good results.

This method could have real life applications and can be proven useful to both researchers and practitioners. More particularly, since creating a new dataset can be expensive, as images need to be collected and correctly annotated, the rationale behind our approach is that if pretrained models on datasets with similar annotated objects are

available, even a few extra training samples can be adequate for the new data and task.

3.2 Evaluation Metrics

Mean Average Precision (mAP) is a popular metric in measuring the accuracy of object detectors like Faster R-CNN, YOLOX etc. It is the basis for many Object Detection competitions and in this thesis, it serves as the basis for evaluation of our models. The metric is based on other popular metrics in Machine Learning, like Precision, Recall and Intersection over Union (IoU), which we will also briefly present.

3.2.1 Precision and Recall

Precision and recall are two important metrics used to evaluate the performance of a classification problem. Precision measures the accuracy of positive predictions made by the system. It calculates the ratio of true positive predictions (correctly predicted positives) to the total number of positive predictions (both true positives and false positives). Precision focuses on how many of the positive predictions are actually correct. The formula for precision is:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Recall, also known as sensitivity or true positive rate, measures the ability of a system to find all the positive instances. It calculates the ratio of true positive predictions to the total number of actual positive instances (true positives and false negatives). Recall focuses on how many of the actual positive instances are correctly identified by the system. The formula for recall is:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Precision and recall are often inversely related. Increasing the precision tends to decrease the recall, and vice versa. This trade-off occurs because increasing the threshold for classifying an instance as positive (thereby reducing false positives) typically leads to a decrease in the number of instances predicted as positive (which can increase false negatives), and vice versa.

To summarize, precision focuses on the quality of positive predictions, while recall focuses on the completeness of positive predictions. Both metrics are valuable and should be considered together to get a comprehensive understanding of the system's performance, especially in scenarios where false positives or false negatives have different implications or costs. These two metrics, as well as the Intersection over Union are the basis for the mean Average Precision, i.e. the basic metric of our experiments.

3.2.2 Intersection over Union

Intersection over Union (IoU) is used to evaluate the performance of object detection by comparing the ground truth bounding box to the predicted bounding box . We use

that to measure how much our predicted boundary overlaps with the ground truth (the real object boundary). Figure 3.1 explains the mathematical calculation of IoU for two boundaries. For object detection models, it is important to predefine an IoU threshold above which the prediction is characterized as a true positive or false positive, as this also distinguishes the different mean Average Precision definitions that are used. An example of a true positive and a false positive detection, if the IoU threshold is set at 0.5, is shown on figure 3.2


$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


Figure 3.1. Intersection over Union definition

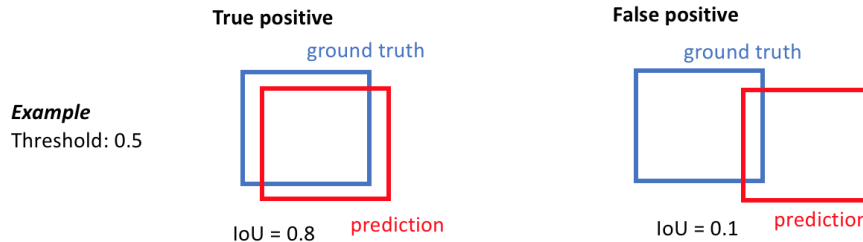


Figure 3.2. True Positive and False Positive definition in relation with IoU

3.2.3 Average Precision and mean Average Precision

After defining precision, recall and IoU, we need to define Average Precision (AP) and mean Average Precision (mAP) which is the basic evaluation metric of our experiments. Average precision computes the average precision value for recall values from 0 to 1, i.e. it measures the precision-recall trade-off at different thresholds for classifying objects. In order to calculate the Average Precision of each category, the first step is to sort the predicted bounding boxes in descending order based on their confidence score. For each predicted bounding box, it is determined whether whether it matches a ground truth bounding box (based on a predefined IoU value, above which the predictions is characterized as true positive). Precision and recall values are calculated at each threshold as the number of true positives, false positives, and false negatives change. The next step

is to plot the precision-recall curve, based on values obtained in step 1. The final step is to compute the average precision as the area under the precision-recall curve. This involves calculating the integral of precision with respect to recall. There are different methods to compute this integral, such as the 11-point interpolation or the area under the curve (AUC) approach. This calculates the Average Precision of each category. Then, mean Average Precision is the mean of Average Precisions of all categories.

The MMDetection toolkit that is presented in 3.4 provides different mAP calculations when the models are evaluated, based on the IoU threshold that is used and some other implementation details. In this thesis, we use the PASCAL VOC [21] definition of average precision, where the prediction is positive if the IoU value is greater than 0.5. If multiple detections of the same object are detected, it counts the first one as a positive while the rest as negatives. In order to calculate AP, an average of 11-point interpolation of precision values for each recall value from 0 to 1 with step equal to 0.1 is calculated. Another definition of mAP is the one defined by the COCO [22] dataset. In COCO mAP, a 101-point interpolated AP definition is used in the calculation. For COCO, AP is the average over multiple IoU (the minimum IoU to consider a positive match). AP@[.5:.95] corresponds to the average AP for IoU from 0.5 to 0.95 with a step size of 0.05. In this thesis, these two mAP values are calculated for all calculated for each experiment. The PASCAL VOC definition is named **mAP50** in the results, whereas the COCO definition is named **mAP50:95**.

3.3 Datasets

3.3.1 DOTA

DOTA [23] is a large-scale dataset for object detection in aerial images. It is widely used to develop and evaluate object detectors in aerial images. The images are collected from different sensors and platforms. Each image is of the size in the range from 800×800 to $20,000 \times 20,000$ pixels and contains objects exhibiting a wide variety of scales, orientations, and shapes. DOTA provides annotations both in oriented and horizontal bounding boxes. For this thesis, the horizontal bounding boxes were used, since the other two datasets that will be evaluated only contained annotations for horizontal bounding boxes.

The team behind DOTA has released three versions of the dataset. DOTAv1 contains 15 categories, with 2,806 images and 188,282 annotated instances. DOTAv1.5 introduced another category and also included annotations of extremely small instances (less than 10 pixels). Finally, the last iteration of the dataset, DOTAv2, collected more Google Earth, GF-2 Satellite, and aerial images and introduced another two categories, for a total of 18 categories. In total there are 11,268 images and 1,793,658 instances in DOTAv2, split into training, validation and test sets. For this thesis, the training and validation sets of DOTAv2 were used, since the test set does not include annotations and is used for the evaluation of the DOTA competition. The training set contains 1,830 images and 268,627 annotations whereas the validation set contains 593 images and 81,048 annotations. For all experiments, the training and validation images were split in patches of size 512×512

pixels. Table 3.1 presents the number of instances for each of the 18 categories for the training and validation sets, where the problem of class imbalance is shown, while figure 3.3 presents annotated examples of the 18 categories taken from the DOTA website.

DOTA is diverse dataset that covers many of the important challenges inherent in aerial imagery such as image and object scale variance, cluttered arrangements, arbitrary orientations and aspect ratios and has category imbalance.

Category	Training Set	Validation Set
plane	8383	2832
baseball diamond	533	273
bridge	2402	546
ground track field	473	186
small vehicle	169268	50062
large vehicle	24570	5371
ship	40552	13801
tennis court	2600	783
basketball court	564	147
storage tank	7372	3175
soccer ball field	420	152
roundabout	631	228
harbor	6455	2443
swimming pool	2379	851
helicopter	652	78
container crane	256	14
airport	305	104
helipad	102	2

Table 3.1. *Number of Objects in DOTAv2*

3.3.2 HRRSD

The first dataset that the pretrained models on DOTAv2 were transferred and evaluated was the High Resolution Remote Sensing Detection (HRRSD) dataset [24]. HRRSD contains 21,761 images acquired from Google Earth and Baidu Map with the spatial resolution from 0.15m to 1.2m. It contains 55,740 object instances and 13 categories of Remote Sensing Imaging objects. 10 of these categories are common with DOTAv2 dataset, while 3 categories are unique to HRRSD and are not present in DOTAv2. The training and test subsets of the dataset are used in this thesis, containing 4,352 and 13,057 images respectively. Table 3.2 presents the number of annotations of each category, while with gray font we present the categories that are not present in DOTAv2 and are not evaluated in our experiments. The annotations in HRRSD are given in horizontal bounding boxes.

As explained in 3.1, we conduct experiments while training the pretrained DOTAv2 models on different percentages of the HRRSD training set. Similarly to DOTAv2, the training set is split in patches of size 512x512 sizes. Table 3.3 contains the number of the full training and test images, the number of the training images when split into



Figure 3.3. Examples of DOTA annotations for each category

patches as well as the number of the patches that are used in the 5 different training experiments.

3.3.3 ITCVD

The final dataset that is being evaluated in this thesis is the ITCVD [25] dataset. ITCVD dataset is a large-scale, well annotated and challenging vehicle detection dataset. The images were taken from an airplane platform which flew over Enschede, The Netherlands, in the height of 330m above the ground. The images are taken in both nadir view and oblique view where the tilt angle is 45 degrees. In total, the dataset contains 173 images, split into 135 images with 23,543 vehicles in the training set and 38 images with 5,545 vehicles in the testing set. Each vehicle in the dataset is manually annotated using a horizontal bounding box.

The pretrained models of DOTAv2 are evaluated in the task of vehicle detection on the new dataset, since vehicle is one of the categories of DOTAv2. For that, we evaluate the models both in the full testing images and also in the testing images split in 512x512 patches, as will be presented in section 4.3.

Category	Training Set	Test Set
bridge	966	2713
airplane	1073	2837
ground track field	809	2446
vehicle	908	2890
parking lot	1042	2861
T junction	934	2729
baseball diamond	797	2495
tennis court	795	2713
basketball court	802	2384
ship	740	2428
crossroad	867	2613
harbor	838	2231
storage tank	839	2852

Table 3.2. Number of Objects in HRRSD

Set	Number of Images
Training Set	4352
Testing Set	13057
Training Set (Patches with objects)	27648
0.5% of training set	138
1% of training set	276
5% of training set	1382
10% of training set	2764
20% of training set	5529

Table 3.3. Number of Images in the different HRRSD datasets

3.4 Training and Evaluation Procedure

MMDetection [26] is an open source object detection toolbox based on PyTorch [27]. It is a part of the [OpenMMLab project](#). For the experiments conducted during this thesis, solely the MMDetection framework was used on a single GPU computer, which can train and evaluate a variety of Object Detector models on different datasets. In addition, MMDetection includes results for the models trained on the COCO [22] dataset, a well known dataset for Object Detection tasks that contain 80 different categories of objects. Some of these results are also presented in chapter 4. Finally, tools from the codebase included in the paper 'Object Detection in Aerial Images: A Large-Scale Benchmark and Challenges' [28] for splitting images for training and translating the DOTA annotations into COCO annotation format, in order to be used with the MMDetection toolbox were used. The same paper also contains several benchmarks for certain object detectors on the DOTA datasets, also presented in chapter 4.

Results and Evaluation

This chapter contains both quantitative and qualitative results from the experiments conducted during this thesis. Section 4.1 presents the results of training the five models presented in chapter 2 in the DOTAv2 dataset, whereas sections 4.2 and 4.3 present the results of applying the technique of transfer learning to the HRRSD and ITCVD datasets.

4.1 Results on DOTAv2

The five models were first trained on the DOTAv2 dataset. The 4 models that were available in the toolbox were available as pretrained in the COCO [22] dataset, while the Faster R-CNN variation model with ConvNeXt backbone was created and trained from scratch. Figure 4.1 presents the training losses of our experiments for the 5 models. Table 4.1 presents the results for the two metrics that were introduced and explained in section 3.2 for all experiments. Results from related experiments from similar studies (referenced in the table) are presented in gray font, while the results of our experiments are presented in black font. For each case, the higher mAP from the five models is shown in bold font.

We observe that the 5 models exhibit similar results when evaluated on the DOTAv2 development set, both when the development set is split and when it is not. The mean Average Precision of the models on the development set when split in patches is high when compared with similar experiments from the past [28], given that DOTAv2 is characterized as a difficult dataset to exhibit good results compared to its predecessors DOTAv1 and DOTAv1.5. The poor results of the models when evaluated on the full images of the development set is due to a number of factors. Firstly, DOTAv2 images have a large size with objects in various relative sizes and the object detectors often miss the smaller sized objects, since part of the algorithms is the rescaling of the input image. Secondly, the algorithms set a number of maximum detections when evaluated that are often surpassed by the big number of annotated objects of the development set images. The solution for detecting objects in the full image is to detect the objects in smaller sized patches of the image and then merge the results together, something that was not explored during the experiments.

In addition, we observe that the two models that were able to achieve the best results in DOTAv2 were YOLOX and Deformable DETR, which are more recent, advanced and

sophisticated in relation to the Faster R-CNN and RetinaNet object detectors.

These five models that were trained on DOTAv2 were the basis for the subsequent results of this thesis, presented in sections 4.2 and 4.3.

Metric	Faster R-CNN	RetinaNet	YOLOX	Deformable DETR	Faster R-CNN ConvNeXt
mAP50:95 COCO [26]	0.374	0.374	0.405	0.445	-
mAP50 COCO [26]	0.581	0.567	0.591	0.632	-
mAP50 DOTAv2 test [28]	0.5071	0.4931	-	-	-
mAP50:95 DOTAv2 dev (patch)	0.3292	0.3019	0.3421	0.3404	0.3094
mAP50 DOTAv2 dev (patch)	0.5367	0.5034	0.5549	0.5515	0.506
mAP50:95 DOTAv2 dev (full)	0.129	0.099	0.099	0.133	0.103
mAP50 DOTAv2 dev (full)	0.198	0.159	0.17	0.228	0.163

Table 4.1. Results of DOTAv2

4.2 Results on HRRSD

4.2.1 Quantitative Results

After training the five models that were studied during this thesis in the DOTAv2 dataset, the trained models are transferred to a new dataset to be evaluated. The first dataset that was evaluated was HRRSD, with the results presented in this section. Six experiments were conducted. The first was to evaluate the models on the new dataset without further training (zero-shot). The next experiments were to further train all models on five different subsets of the HRRSD train set, as explained in section 3.1. The number of 512x512 sized patches that are part of each of these five experiments were presented in table 3.3.

Tables 4.2 and 4.3 present the mean Average Precisions with the two different definitions for these experiments. We observe that although the results of the zero-shot case are poor compared to the results that these five models obtained on DOTAv2, further training them even on a very small subset of the training set can significantly increase the Average Precision. The results of these tables are also presented in figures 4.2 and 4.3, showing for the 5 models the increase in mean Average Precision with the increase of the training set size. As far as the models that were used, we observe that the Faster R-CNN and the YOLOX model were able to achieve the greater mAP. This is not the case of Deformable DETR, which although had the second best results in the baseline dataset DOTAv2, it was not able to transfer to the new dataset accordingly.

Tables 4.4 through 4.9 include class-wise Average Precisions for each of the ten common categories of the HRRSD and DOTAv2 datasets and for each of the six experiments. For the case of zero-shot learning, very few categories have high average precision (like storage tanks and tennis courts), whereas the remaining categories are hard to detect without further training. However, when trained on 20% of the dataset, most of the categories increase their average precision results significantly. There are still a few categories that are hard to detect, like bridges or basketball courts and baseball diamonds, that will be discussed in the next section with the qualitative results. In addition, we observe that different models display the best behavior in different categories for each experiment. For example, although Faster R-CNN has the best Average Precision in most categories in the case of few-shot learning, bridges are better detected by the Deformable DETR algorithm and certain categories are better detected by YOLOX.

4.2.2 Qualitative Results

To further investigate the results and the behavior of the models in the HRRSD dataset, we select six images from the test set, presented along with their ground truth annotations in figures 4.4, 4.5, 4.6, 4.7, 4.8 and 4.9. The first and very important observation from these images is that the annotations of the HRRSD dataset are not complete. More specifically, we observe that in the larger images of cases 1, 3, 5 and 6 that contain larger objects, like basketball courts, bridges or storage tanks, the smaller images that are present (in this case vehicles) were not annotated. In contrast, the small image of case 2 has annotated vehicles. In addition, images of cases 1, 3 and 6 contain the 'hard' categories to detect (bridge, baseball diamond and basketball court), as presented in the class-wise results.

Figures 4.10, 4.11, 4.12, 4.13, 4.14 and 4.15 present the detections of the model with the best overall precision (Faster R-CNN) for each of the six experiments (zero-shot and few-shot with different training sizes) and for each of the six cases that were presented before. Some important observations per case are listed below:

- In case 1, we observe that for the case of zero-shot learning, several vehicles are detected. As we explained earlier, these vehicles are not annotated in the ground truth annotations and thus they are considered false positives in the calculation of average precision. In addition, we see the difficulty of the models to distinguish between categories with similar features as the ground track field is mistakenly detected as a tennis court. Finally, only half the models are able to detect the presence of basketball courts.
- In case 2, vehicles are detected with ease by all models that are trained on the new dataset, while in the case of zero-shot learning, few of the vehicles are missed. In general, vehicles are considered an easy category, only in smaller images that do not contain other annotated objects.
- Case 3 contains one of the harder categories to detect, bridges. Bridges have several different aspect ratios and it is also difficult to concretely define its extents. We

can see that in all models, there are additional 'bridges' detected, which are in fact false positives. However the actual bridge is detected in the case of few-shot learning, but not in the case of zero-shot learning

- Case 4 presents a view of an airport, with main parked airplanes. Planes are detected easily in most of the models that are evaluated and very few are missed by the detectors and are considered as false negatives in the average precision calculation. We can see that again the presence of vehicles in the original image is missed since they are not annotated, and they are subsequently not detected either.
- Case 5 contains several annotated storage tanks. We can observe though that the smaller storage tanks that are apparent in the original image are not annotated, marking another problem in the HRRSD dataset. The large storage tanks are easily detected by the model in both cases of zero-shot and few-shot learning. In addition, for the first three cases (zero-shot learning and few-shot learning with 0.5% and 1% of the training set), smaller storage tanks are detected as well and are thus considered false positives in the results.
- In case 6, the original annotated image contains the categories of ground track field, baseball diamond and tennis court. Again the presence of vehicles is missed in the annotations of the dataset, and although some are detected in the case of zero-shot, they are considered false positives. We can see that the ground track field is now easily detected and not confused with a tennis court, like in case 1. However the tennis courts and baseball diamonds are missed and counted as false negatives in all of the models with very few exceptions. Another important observation is that in the case of zero-shot learning, a soccer ball field is detected. This is one of the categories that are present in the DOTA_{v2} dataset but not in the HRRSD dataset. Thus the model that is only trained on DOTA_{v2} is able to detect it. This does not affect the calculation of the mean Average Precision of this case, since it is calculated only for categories present in HRRSD.

Training mode	Faster R-CNN	RetinaNet	YOLOX	Deformable DETR	Faster R-CNN ConvNeXt
Zero-shot	0.264	0.257	0.360	0.256	0.262
0.5% of train set	0.490	0.444	0.490	0.397	0.417
1% of train set	0.626	0.564	0.557	0.499	0.484
5% of train set	0.717	0.631	0.550	0.654	0.610
10% of train set	0.737	0.648	0.568	0.674	0.598
20% of train set	0.751	0.686	0.780	0.682	0.678

Table 4.2. *mAP₅₀ Results on HRRSD*

Training mode	Faster R-CNN	RetinaNet	YOLOX	Deformable DETR	Faster R-CNN ConvNeXt
Zero-shot	0.163	0.159	0.213	0.161	0.158
0.5% of train set	0.278	0.242	0.281	0.202	0.217
1% of train set	0.356	0.303	0.322	0.275	0.255
5% of train set	0.443	0.320	0.282	0.330	0.364
10% of train set	0.464	0.343	0.287	0.349	0.323
20% of train set	0.488	0.373	0.452	0.353	0.418

Table 4.3. *mAP50:95 Results on HRRSD*

Training mode	Faster R-CNN	RetinaNet	YOLOX	Deformable DETR	Faster R-CNN ConvNeXt
plane	0.129	0.137	0.217	0.157	0.148
baseball diamond	0.035	0.013	0.027	0.019	0.021
bridge	0.009	0.006	0.034	0.014	0.010
ground track field	0.177	0.123	0.183	0.070	0.146
small vehicle	0.004	0.006	0.088	0.008	0.001
ship	0.013	0.006	0.111	0.063	0.012
tennis court	0.598	0.571	0.620	0.556	0.594
vasketball court	0.164	0.152	0.166	0.160	0.135
storage tank	0.484	0.529	0.550	0.537	0.478
harbor	0.021	0.043	0.134	0.027	0.035

Table 4.4. *Classwise AP50:95 on HRRSD - Zero-shot*

4.3 Results on ITCVD

4.3.1 Quantitative Results

The 5 models that were pre-trained on the DOTA2 are evaluated on the ITCVD test set that was presented in section 3.3.3, without further training. Tables 4.10 and 4.11 present the results for the two evaluation metrics and for both evaluation modes. Again, in bold font is the model with the best mAP for each evaluation mode. We can observe that again YOLOX is the model with the best results on this transfer learning task. Since for this dataset only one category is detected, Average Precision and mean Average Precision are the same. On one case we split the test set into patches of size 512x512 pixels before evaluating the performance of the models, while on the other case we evaluate the models on the full image. We detect very high mean Average Precision values for the models evaluated on the patched images, whereas the results on the experiments on the full images are rather poor. The reasons are similar to the ones presented in section 4.1, namely the relative size of the vehicles to the full image that is re-scaled before passing the evaluation pipeline and the fact that we have a maximum number of detections set in all models.

Training mode	Faster R-CNN	RetinaNet	YOLOX	Deformable DETR	Faster R-CNN ConvNeXt
plane	0.385	0.323	0.563	0.313	0.424
baseball diamond	0.074	0.079	0.088	0.055	0.017
bridge	0.075	0.044	0.021	0.066	0.031
ground track field	0.397	0.355	0.382	0.214	0.343
small vehicle	0.230	0.247	0.229	0.121	0.226
ship	0.264	0.112	0.267	0.213	0.142
tennis court	0.541	0.543	0.395	0.415	0.384
vasketball court	0.167	0.128	0.141	0.122	0.075
storage tank	0.565	0.553	0.647	0.432	0.467
harbor	0.078	0.041	0.073	0.069	0.062

Table 4.5. Classwise AP50:95 on HRRSD - 0.5% of training set

Training mode	Faster R-CNN	RetinaNet	YOLOX	Deformable DETR	Faster R-CNN ConvNeXt
plane	0.444	0.454	0.550	0.384	0.346
baseball diamond	0.164	0.104	0.112	0.131	0.051
bridge	0.119	0.073	0.065	0.088	0.059
ground track field	0.493	0.360	0.475	0.370	0.382
small vehicle	0.299	0.363	0.205	0.207	0.110
ship	0.400	0.316	0.334	0.286	0.247
tennis court	0.563	0.456	0.438	0.445	0.527
basketball court	0.185	0.098	0.051	0.110	0.118
storage tank	0.572	0.525	0.580	0.472	0.430
harbor	0.318	0.276	0.409	0.251	0.279

Table 4.6. Classwise AP50:95 on HRRSD - 1% of training set

4.3.2 Qualitative Results

Finally, we present some qualitative results to demonstrate the observations of the previous section. To that end, two images from the test set are selected, which are being presented in figures 4.16 and 4.17. The two images also include a box that corresponds to the patch that will be evaluated along with the full image. One image is taken in nadir view, similarly with the ones from the original dataset, while the other one is taken in oblique view with an angle of 45 degrees. The images are evaluated with the 5 models and figures 4.18 and 4.19 contain the detections of vehicles for each of the 5 models. We can observe that all models exhibit very good results, as all the vehicles in the images are correctly detected, even for the image that is taken in an oblique view. This is not the case however when the models are evaluated in the full image, as demonstrated in figures 4.20 and 4.21, where the Faster R-CNN algorithm is evaluated on the full images. We can observe that very few of the many vehicles that are present in the original images are detected, and the ones that were correctly detected in the patches are not identified. That explains the low Average Precision of the experiments that is presented in tables 4.10 and 4.11.

Training mode	Faster R-CNN	RetinaNet	YOLOX	Deformable DETR	Faster R-CNN ConvNeXt
plane	0.581	0.475	0.475	0.473	0.520
baseball diamond	0.272	0.176	0.159	0.253	0.233
bridge	0.211	0.137	0.084	0.236	0.146
ground track field	0.580	0.458	0.522	0.519	0.553
small vehicle	0.493	0.294	0.152	0.181	0.306
ship	0.414	0.257	0.201	0.280	0.269
tennis court	0.615	0.448	0.245	0.438	0.491
basketball court	0.238	0.118	0.026	0.142	0.098
storage tank	0.607	0.514	0.619	0.485	0.588
harbor	0.415	0.325	0.342	0.297	0.431

Table 4.7. Classwise AP50:95 on HRRSD - 5% of training set

Training mode	Faster R-CNN	RetinaNet	YOLOX	Deformable DETR	Faster R-CNN ConvNeXt
plane	0.607	0.503	0.487	0.514	0.487
baseball diamond	0.255	0.249	0.154	0.275	0.217
bridge	0.212	0.131	0.083	0.240	0.093
ground track field	0.607	0.486	0.509	0.515	0.517
small vehicle	0.484	0.383	0.188	0.199	0.340
ship	0.444	0.261	0.198	0.290	0.166
tennis court	0.601	0.432	0.261	0.438	0.488
basketball court	0.286	0.147	0.043	0.183	0.154
storage tank	0.648	0.493	0.564	0.495	0.437
harbor	0.494	0.344	0.379	0.342	0.328

Table 4.8. Classwise AP50:95 on HRRSD - 10% of training set

Training mode	Faster R-CNN	RetinaNet	YOLOX	Deformable DETR	Faster R-CNN ConvNeXt
plane	0.600	0.521	0.607	0.501	0.559
baseball diamond	0.301	0.216	0.389	0.330	0.261
bridge	0.233	0.201	0.212	0.253	0.194
ground track field	0.649	0.527	0.642	0.558	0.587
small vehicle	0.525	0.403	0.421	0.137	0.374
ship	0.439	0.311	0.358	0.267	0.377
tennis court	0.658	0.489	0.533	0.431	0.566
basketball court	0.298	0.199	0.254	0.200	0.236
storage tank	0.643	0.489	0.684	0.488	0.557
harbor	0.537	0.373	0.419	0.363	0.468

Table 4.9. Classwise AP50:95 on HRRSD - 20% of training set

Evaluation mode	Faster R-CNN	RetinaNet	YOLOX	Deformable DETR	Faster R-CNN ConvNeXt
Zero-shot patch	0.731	0.749	0.871	0.826	0.725
Zero-shot full	0.108	0.123	0.014	0.014	0.105

Table 4.10. AP50 Results on ITCVD

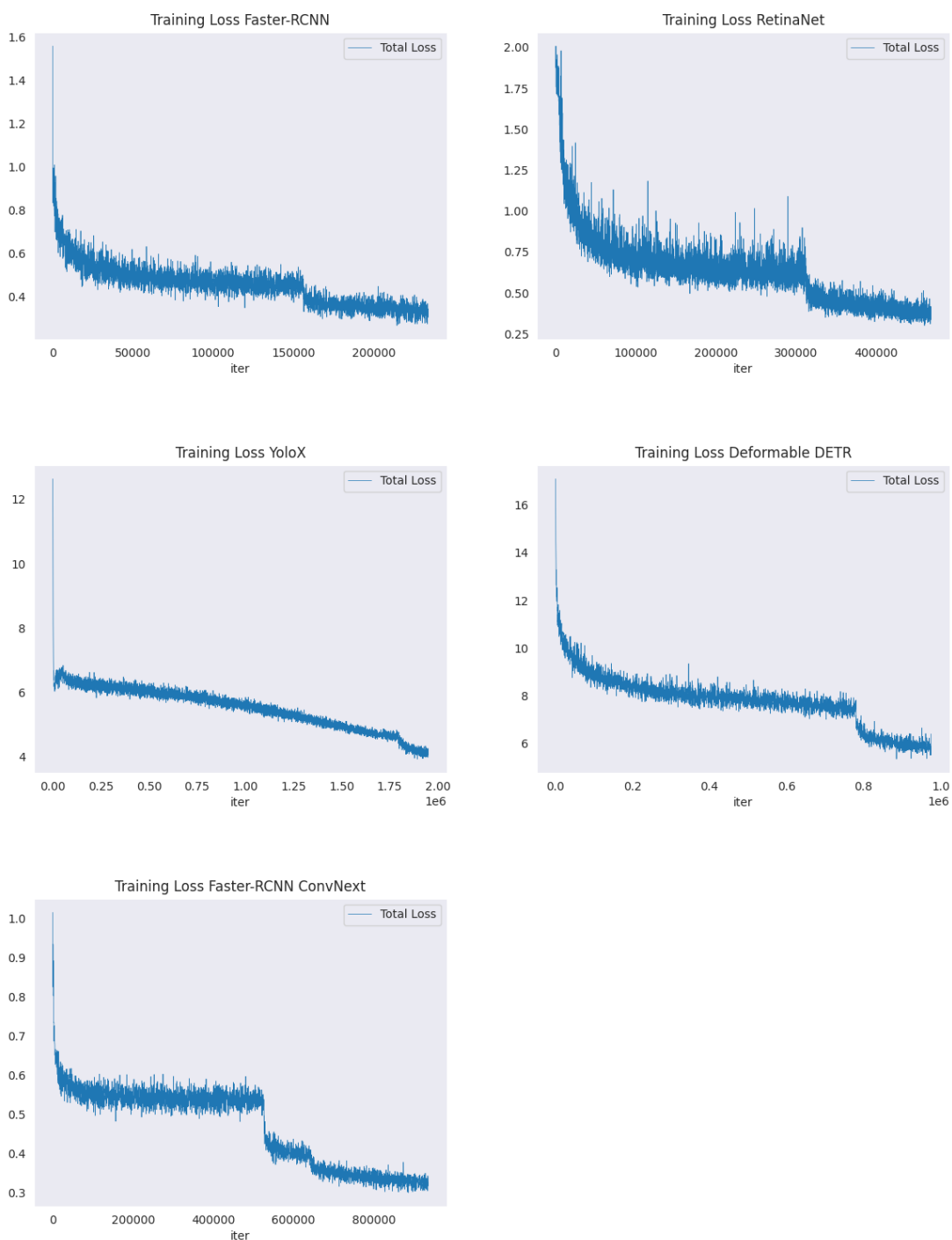


Figure 4.1. Training Loss in DOTAv2

Evaluation mode	Faster R-CNN	RetinaNet	YOLOX	Deformable DETR	Faster R-CNN ConvNeXt
Zero-shot patch	0.439	0.440	0.506	0.499	0.432
Zero-shot full	0.041	0.042	0.003	0.003	0.036

Table 4.11. AP50:95 Results on ITCVD

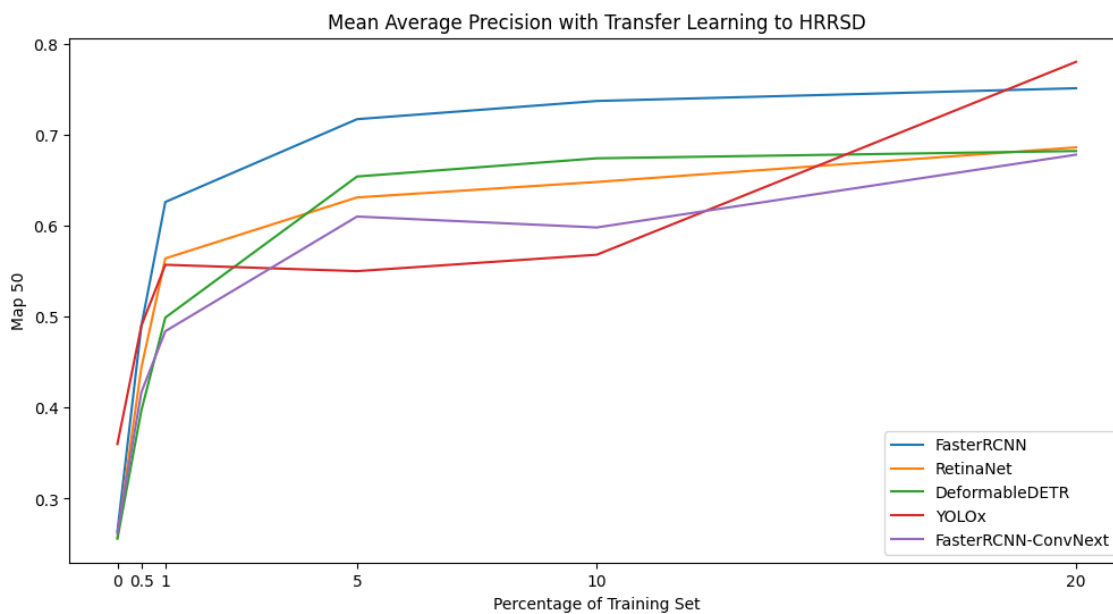


Figure 4.2. Transfer Learning mAP50 curve for different percentages of training set

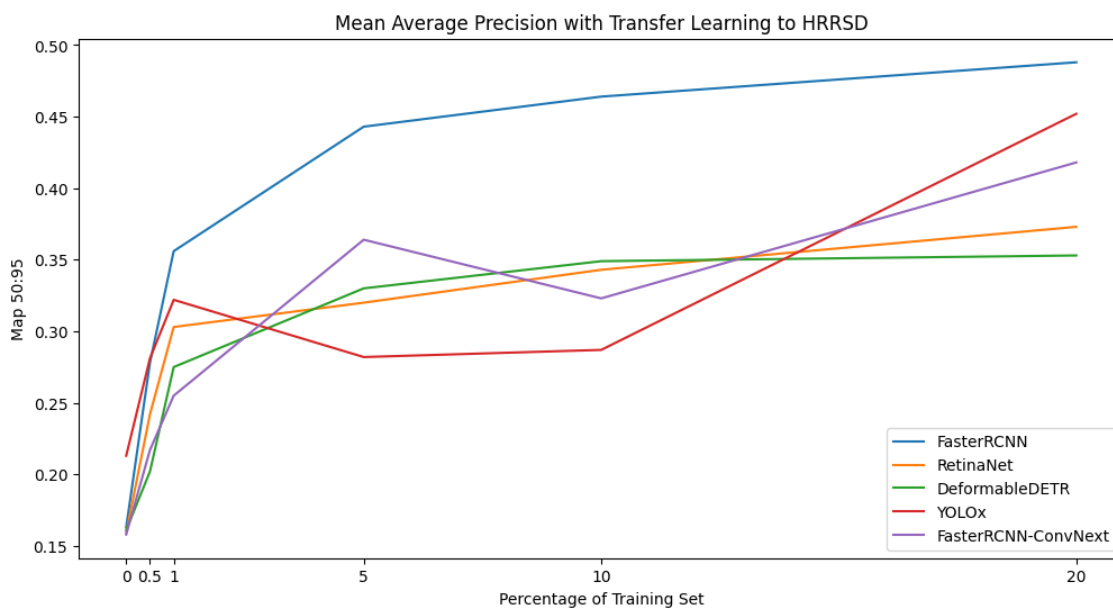


Figure 4.3. Transfer Learning mAP50:95 curve for different percentages of training set



Figure 4.4. Test Image from HRRSD with ground truth annotations - Case 1



Figure 4.5. Test Image from HRRSD with ground truth annotations - Case 2

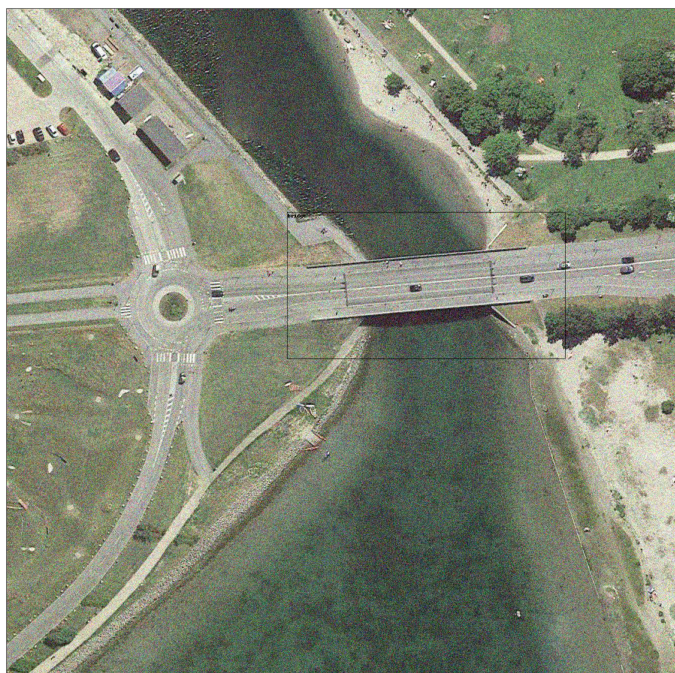


Figure 4.6. Test Image from HRRSD with ground truth annotations - Case 3



Figure 4.7. Test Image from HRRSD with ground truth annotations - Case 4

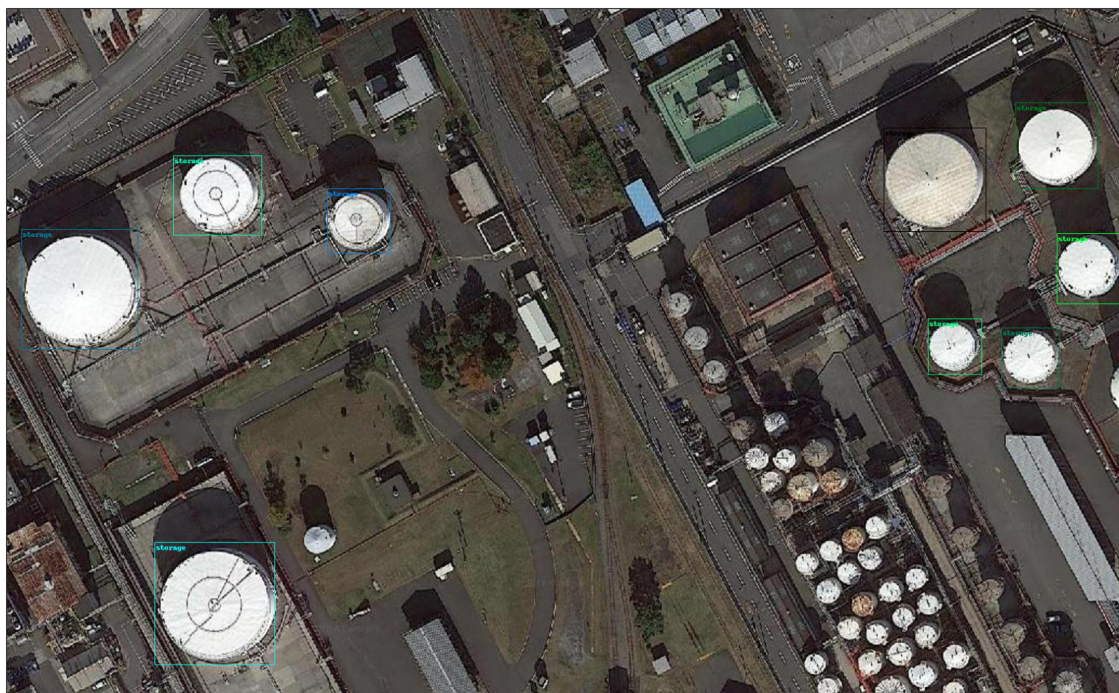


Figure 4.8. Test Image from HRRSD with ground truth annotations - Case 5

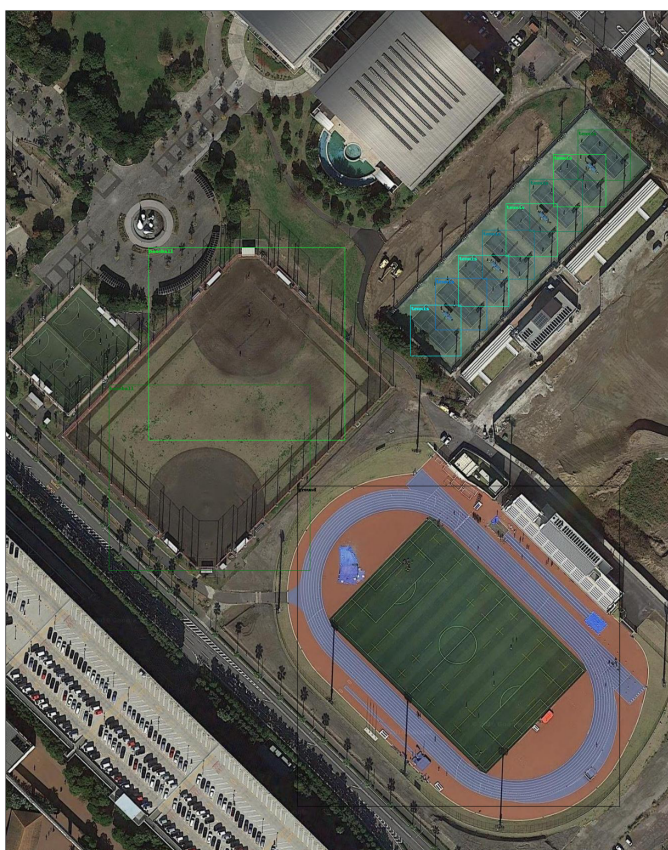


Figure 4.9. Test Image from HRRSD with ground truth annotations - Case 6

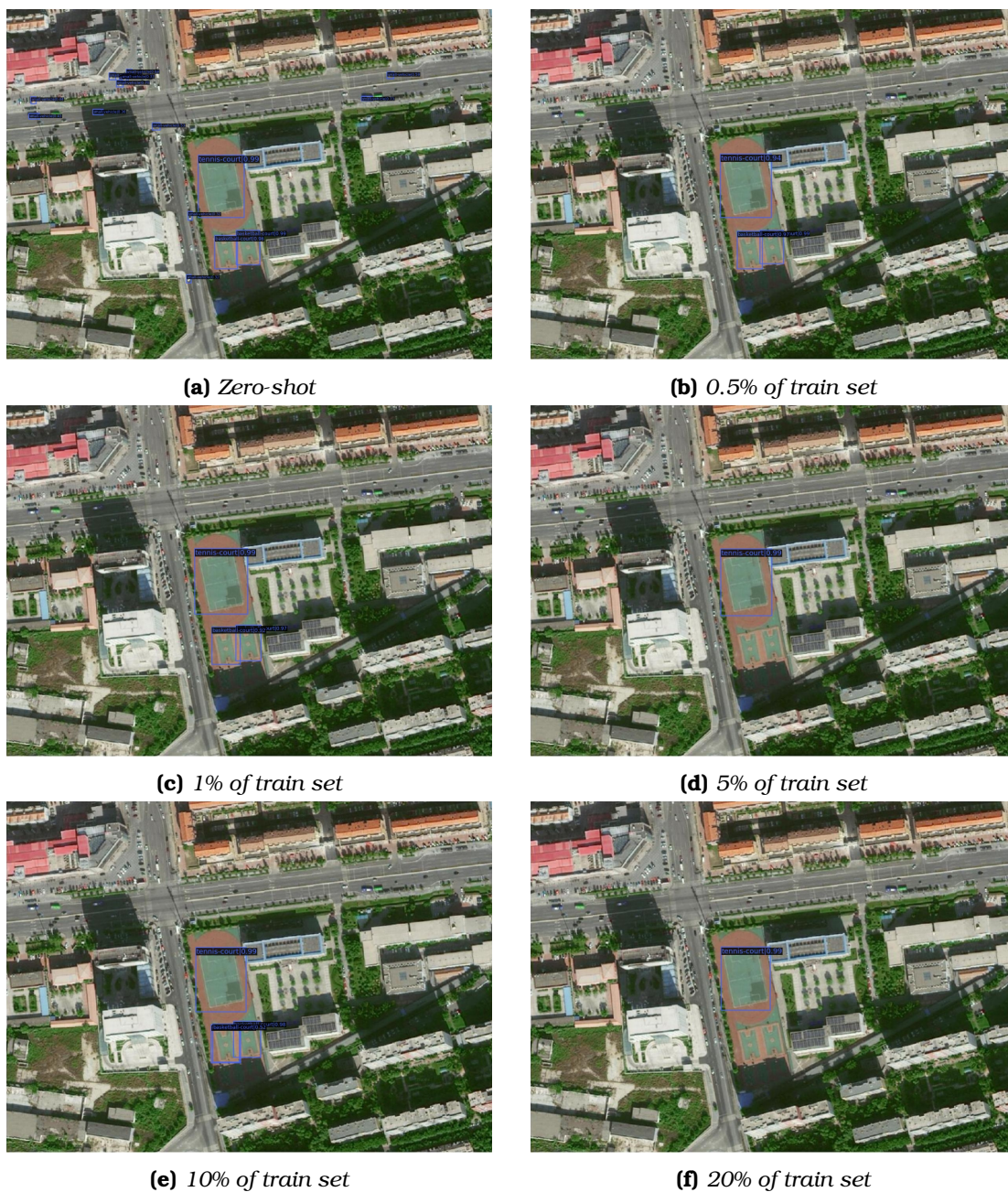


Figure 4.10. Object Detections in HRRSD - Case 1

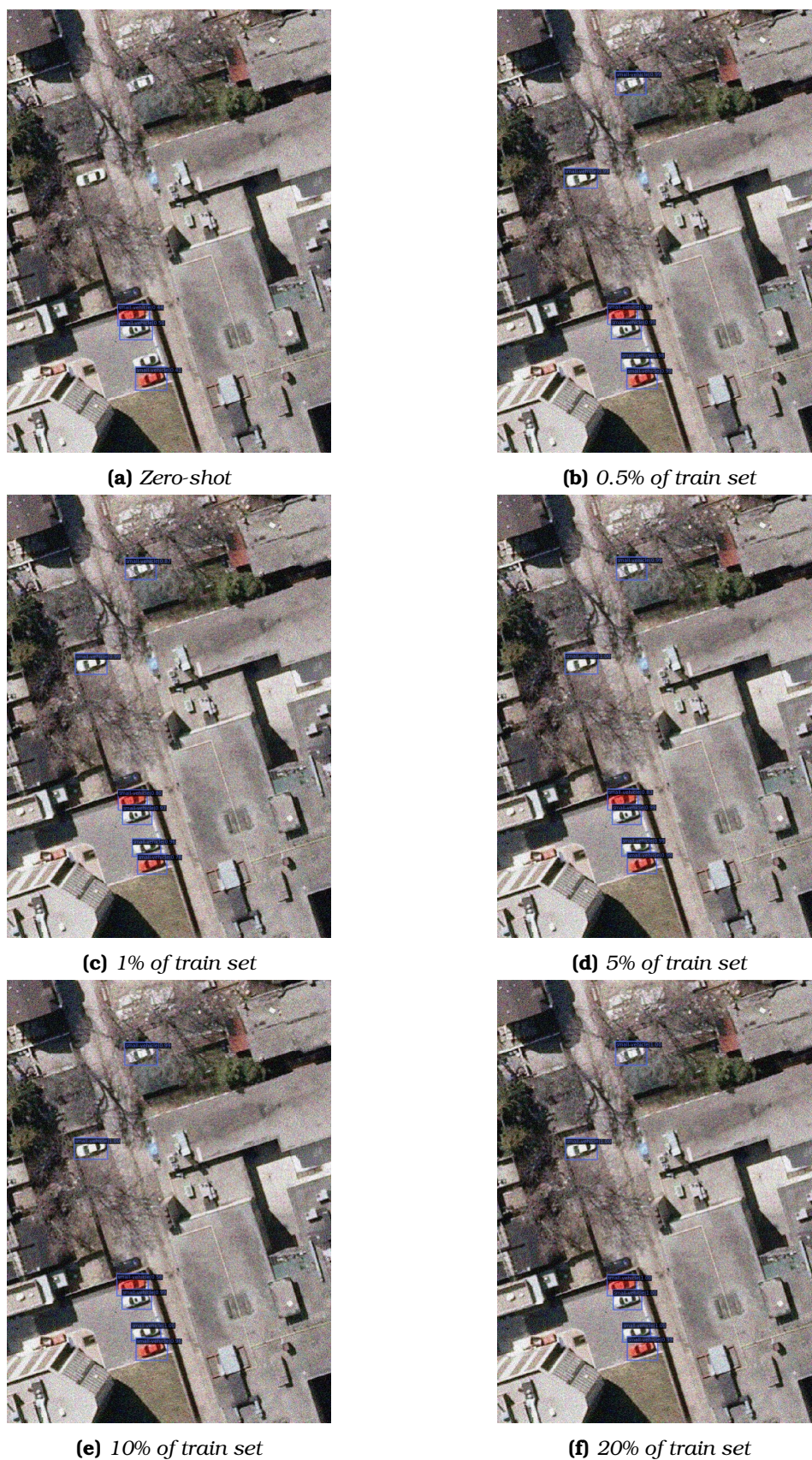


Figure 4.11. Object Detections in HRRSD - Case 2

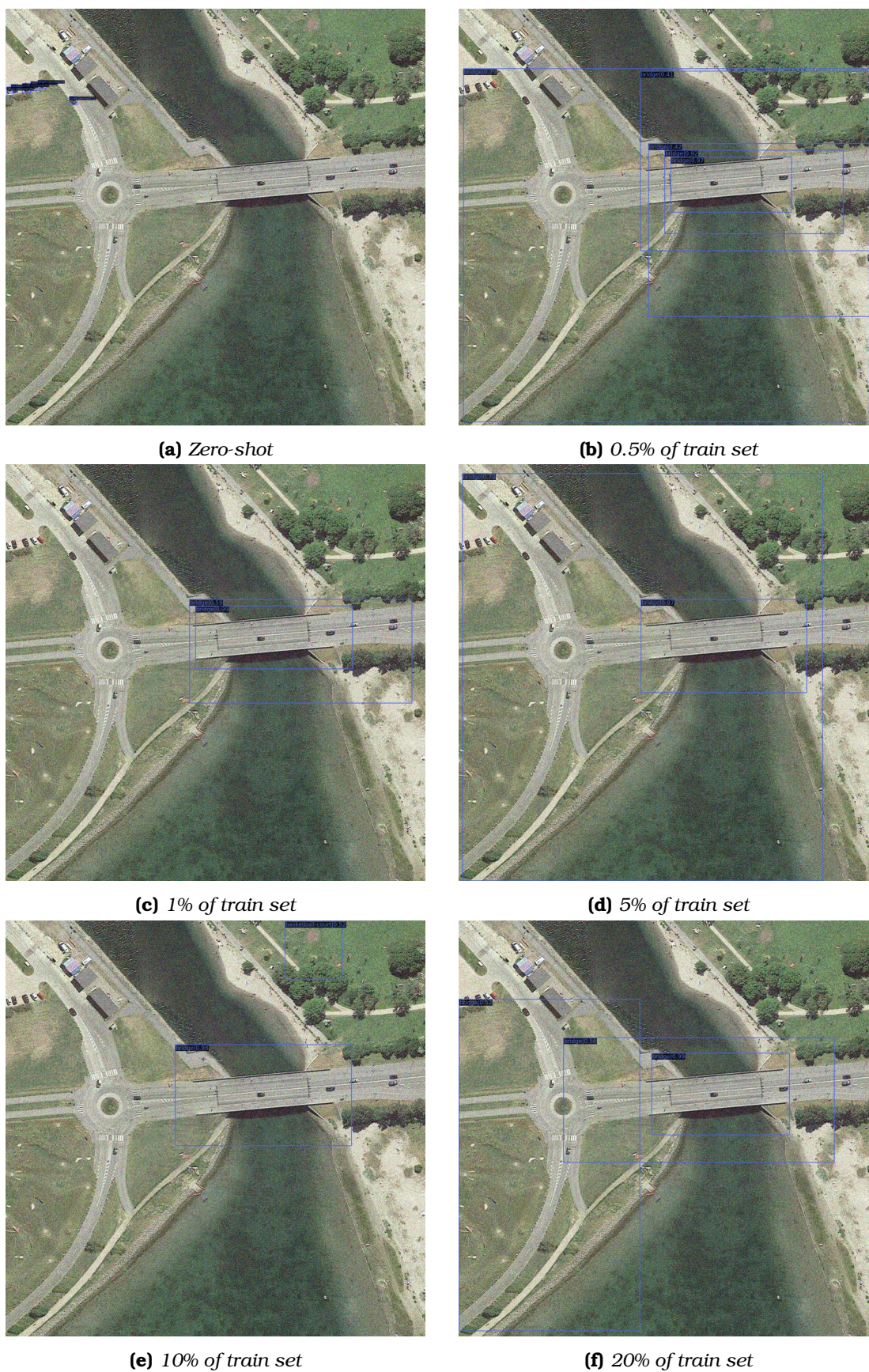


Figure 4.12. Object Detections in HRRSD - Case 3

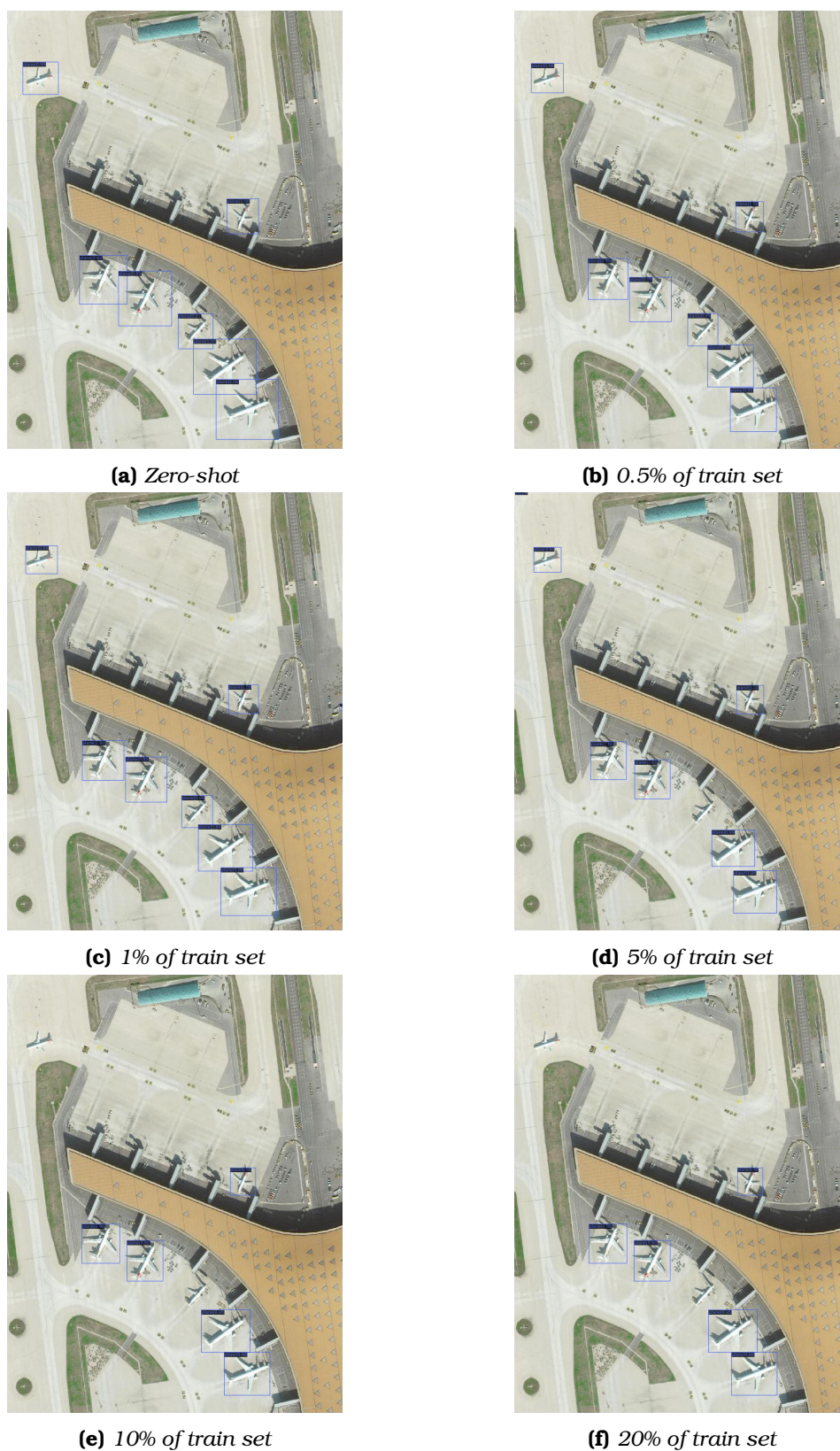


Figure 4.13. Object Detections in HRRSD - Case 4

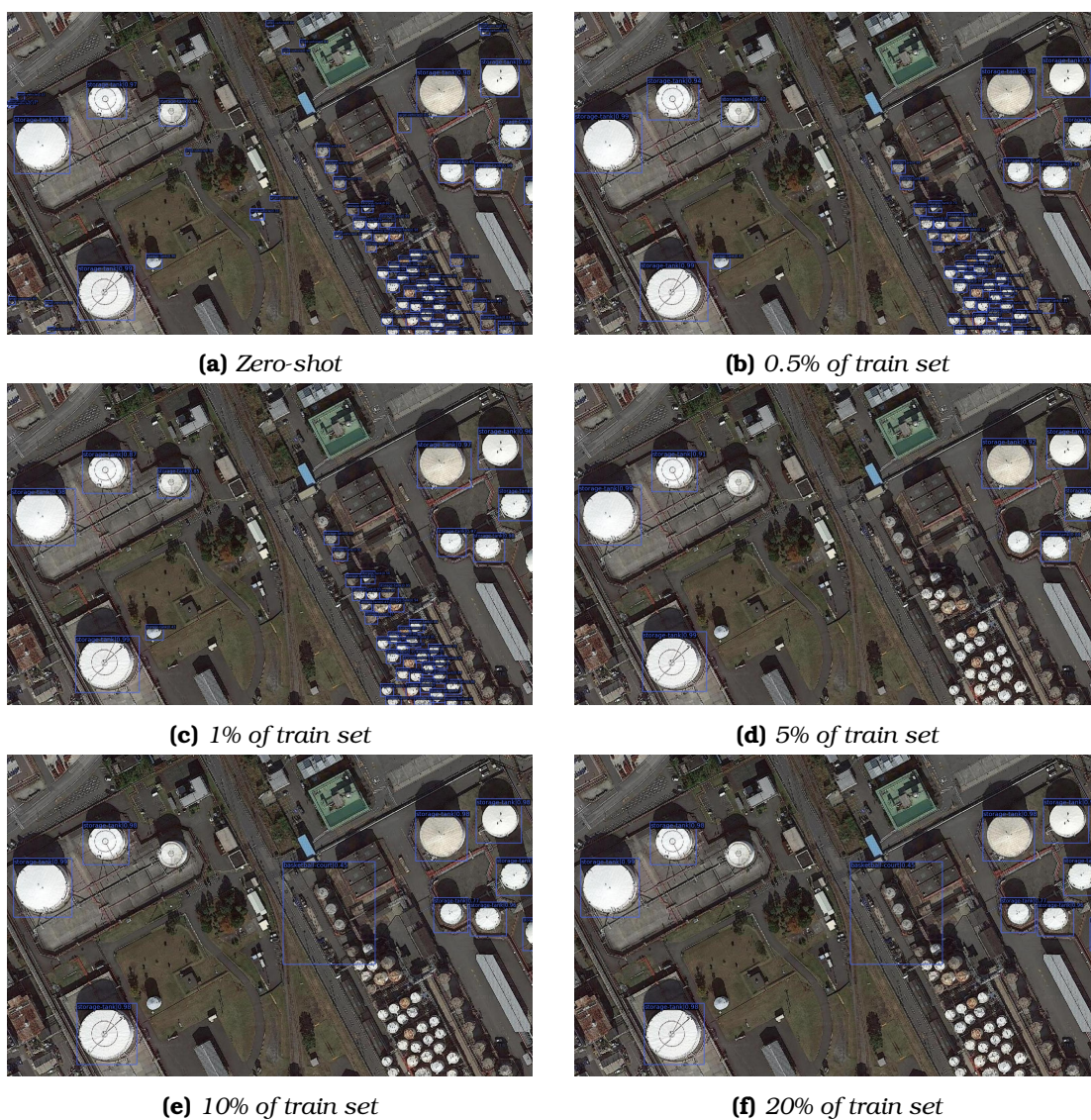
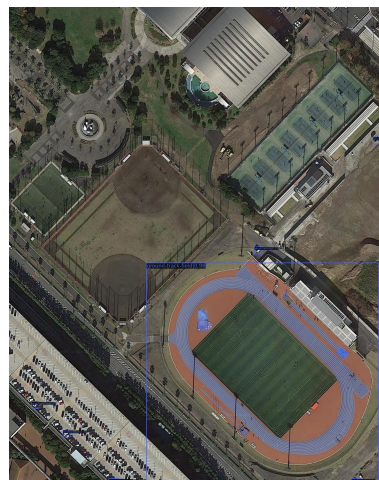


Figure 4.14. Object Detections in HRRSD - Case 5



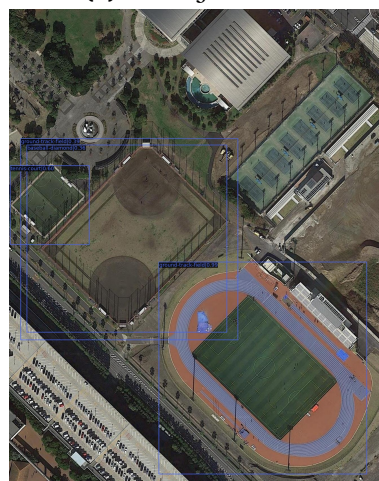
(a) Zero-shot



(b) 0.5% of train set



(c) 1% of train set



(d) 5% of train set



(e) 10% of train set



(f) 20% of train set

Figure 4.15. Object Detections in HRRSD - Case 6



Figure 4.16. *Test Image from ITCVD - Case 1*



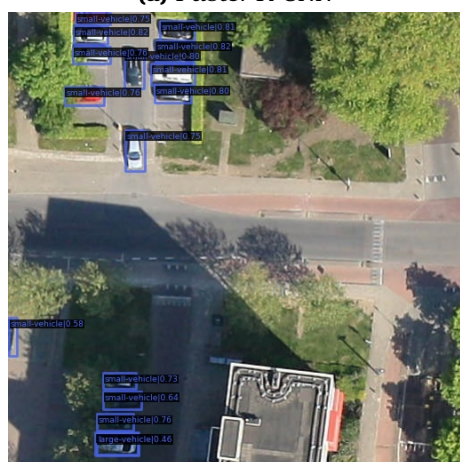
Figure 4.17. *Test Image from ITCVD - Case 2*



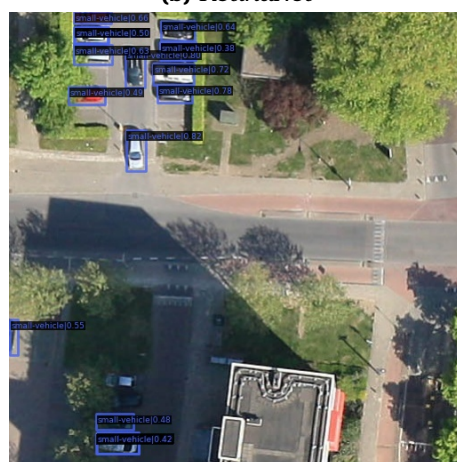
(a) *Faster R-CNN*



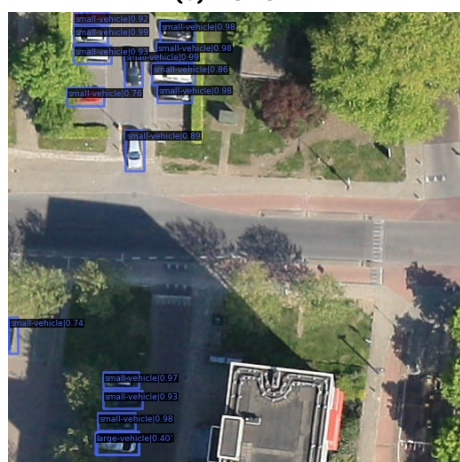
(b) *RetinaNet*



(c) *YOLOX*



(d) *Deformable DETR*



(e) *Faster R-CNN ConvNeXt*

Figure 4.18. Zero-shot Detections in ITCVD patch - Case 1

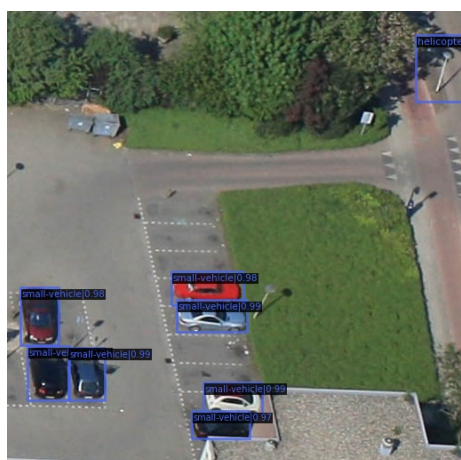
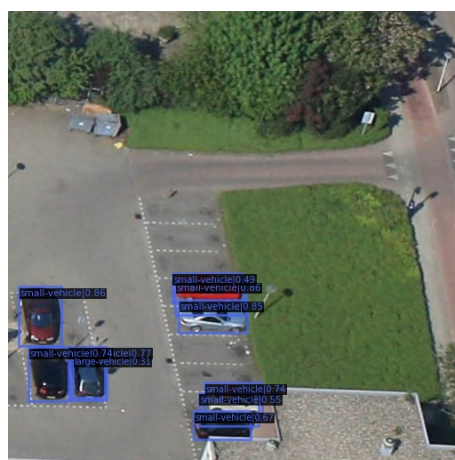
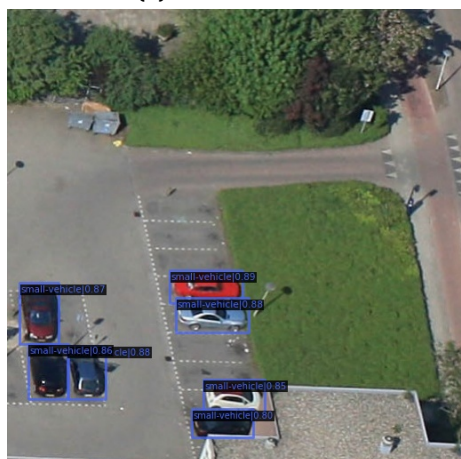
(a) *Faster R-CNN*(b) *RetinaNet*(c) *YOLOX*(d) *Deformable DETR*(e) *Faster R-CNN ConvNeXt***Figure 4.19.** Zero-shot Detections in ITCVD patch - Case 2



Figure 4.20. Zero-shot Detections in ITCVD full image with Faster R-CNN - Case 1



Figure 4.21. Zero-shot Detections in ITCVD full image with Faster R-CNN - Case 2

Chapter 5

Conclusions

This master thesis included a thorough investigation of the topic of Object Detection in Remote Sensing Imagery and the technique of Transfer Learning of trained object detectors in different datasets. For this purpose, five state-of-the-art Object Detectors were presented, analyzed and trained in a large publicly available dataset (DOTAv2), presenting the results of each of the models and evaluating their ability to detect several object categories. The next step was to evaluate these five detectors in two additional datasets (HRRSD and ITCVD), both without further training (zero-shot transfer learning) and with further training on various small subsets of the new datasets (few-shot transfer learning). The results were presented and key observations were documented. Several conclusions can be drawn from these experiments that were presented in chapters 3 and 4 and are also summarized below:

- All object detectors that were used in the experiments and were trained on DOTAv2 displayed a good performance in detecting the categories of DOTAv2, when compared with similar projects from the past. The mean Average Precision of the detectors on the development set, when split into smaller patches, is equal or greater than the mean Average Precision of the test set found in similar projects [28]. This was not the case when mean Average Precision was calculated for detections on the full images of the development set, due to a number of factors, including the various size of objects in the images and the inability of the detectors to identify small objects because of re-scaling of the images in the evaluation pipeline. There are other techniques to identify objects in the full image by detecting the objects in patches and then merging the results into the full image that can be investigated and evaluated in the future.
- The trained object detectors were evaluated in a new and similar dataset (HRRSD) in two different modes. First, the detectors were evaluated without further training them on the new dataset and secondly the detectors were evaluated after being trained in small subsets of the new dataset. The new dataset contained 10 common categories with DOTAv2 and the evaluation of the model's ability to detect objects was restricted only on these common objects. For the case of zero-shot object detection, the mean Average Precision of the detections was very low, compared to similar projects. A key reason for that was that the new dataset was problematic and

many annotations were missed in large-scale images. More specifically, in images that contain large objects like bridges or ground track fields, the smaller objects that may be present are not annotated. Thus, these items might be detected by the zero-shot object detection pipeline and be counted as false positives. When the object detectors were trained even on very few images of the new dataset (e.g. 1% of the training set), the mean Average Precision increases significantly. This lead to the realisation than even a few extra images can help when trying to transfer information from previously learned tasks and datasets to new tasks and datasets.

- Six different cases from the HRRSD test set containing different categories were presented and evaluated. Some of the important observations were that there are categories that are often mixed because they contain similar features and the corresponding objects have the same aspect ratio, like ground track field, tennis courts and baseball diamonds. In addition, there are categories with poor average precision results due to the difficulty to define the extent of the bounding box, like bridges that are shown in one of the cases displayed. Other categories like storage tanks, vehicles and planes are detected with very high precision in the examples shown.
- When the object detectors were evaluated in a dataset that contained only one of the original categories of DOTAv2 (ITCVD that includes only annotated vehicles), the detectors were able to identify the vehicles without further training and with very high precision. Since the ITCVD dataset contained aerial images in both nadir and oblique view, this is a significant achievement given that the original detectors were trained only on images in nadir view that are present in DOTAv2. Thus, we can conclude that transfer learning of the trained detectors for this type of tasks is a real strength that the original detectors possess.

Bibliography

- [1] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *CoRR*, vol. abs/1905.05055, 2019.
- [2] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, pp. I-I, 2001.
- [3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893 vol. 1, 2005.
- [4] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.
- [5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, 2016.
- [6] R. B. Girshick, "Fast R-CNN," *CoRR*, vol. abs/1504.08083, 2015.
- [7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Jun 2017.
- [8] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015.
- [9] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017.
- [10] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, 2020.
- [11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.
- [12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

- [13] J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” 2016.
- [14] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” 2018.
- [15] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOX: Exceeding yolo series in 2021,” *arXiv preprint arXiv:2107.08430*, 2021.
- [16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2017.
- [17] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable detr: Deformable transformers for end-to-end object detection,” in *International Conference on Learning Representations*, 2021.
- [18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” 2021.
- [19] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” 2021.
- [20] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A convnet for the 2020s,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [21] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge.,” *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.
- [22] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, “Microsoft coco: Common objects in context,” 2014. cite arxiv:1405.0312.
- [23] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, “Dota: A large-scale dataset for object detection in aerial images,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [24] Y. Zhang, Y. Yuan, Y. Feng, and X. Lu, “Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5535–5548, 2019.
- [25] M. Y. Yang, W. Liao, X. Li, and B. Rosenhahn, “Vehicle detection in aerial images,” 2018.
- [26] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, “MMDetection: Open mmlab detection toolbox and benchmark,” *arXiv preprint arXiv:1906.07155*, 2019.

- [27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.
- [28] J. Ding, N. Xue, G.-S. Xia, X. Bai, W. Yang, M. Yang, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, “Object detection in aerial images: A large-scale benchmark and challenges,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

List of Abbreviations

ML	Machine Learning
DL	Deep Learning
CV	Computer Vision
TL	Transfer Learning
OD	Object Detection
CNN	Convolutional Neural Network
GPU	Graphical Processing Unit
mAP	mean Average Precision
IoU	Intersection over Union
DOTA	Dataset for Object deTectioN in Aerial images
HRRSD	High Resolution Remote Sensing Detection
COCO	Common Objects in COntext
YOLO	You Only Look Once
DETR	DEtection TRansformer
RPN	Region Proposal Network
NMS	Non Maximum Suppression
FPN	Feature Pyramid Network
DCN	Deformable Convolutional Network