

ONE-STAGE ORIENTED OBJECT DETECTION IN REMOTE SENSING IMAGES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

By
Atakan Serbes
March 2022

ONE-STAGE ORIENTED OBJECT DETECTION IN REMOTE
SENSING IMAGES

By Atakan Serbes

March 2022

We certify that we have read this thesis and that in our opinion it is fully adequate,
in scope and in quality, as a thesis for the degree of Master of Science.

Selim Aksoy(Advisor)

Shervin Rahimzadeh Arashloo

Ramazan Gökberk Cinbiş

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

ONE-STAGE ORIENTED OBJECT DETECTION IN REMOTE SENSING IMAGES

Atakan Serbes
M.S. in Computer Engineering
Advisor: Selim Aksoy
March 2022

Advances in technology resulted in an enormous amount of information collected from high technology satellites and aircraft sensors. These high-resolution images obtained by the said platforms enabled humans to understand the Earth better. The first studies in the field of remote sensing image analysis have focused on image classification, which studies the understanding of scenes, terrain, and vast fields. Recently object detection and image segmentation on very high-resolution electro-optical images have become popular.

Modern object detectors have two types of architecture. Two-stage object detectors screen probable object locations with a region proposal mechanism and regress over these preselected regions to classify objects. On the other hand, one-stage object detectors perform global regression and classification over the image without any preselection. The region proposal mechanism improves the accuracy but slows the detection speed, making one-stage object detectors better for real-time detection, which have lower accuracy. Publicly available data sets generally provide horizontal bounding box labels. However, remote sensing images are dominated by small, congested, or large aspect ratio objects, which is suitable for oriented bounding boxes.

We study one-stage object detection in one of the most extensive remote sensing object detection data sets (DOTA), which includes fifteen object classes. Our one-stage object detectors are based on one of the most prominent one-stage object detector architectures (YOLO). Horizontal and oriented bounding box labels are used to obtain results to show that the oriented object detection setting has a surpassing performance over axis-aligned object detection setting in general and also to show that one-stage object detection can have competitive results

compared to two-stage object detection methods. Our oriented object detection approach converts angle regression into a form of classification and is easily applicable to the one-stage detectors. Our experiments show that rotated object detection based on angular classification performs better than horizontal object detector. It also shows 3.07% improvement over published two-stage methods and 12.84% improvement over published anchor-free methods on the validation data set.



Keywords: Remote sensing image analysis, deep learning, oriented object detection, one-stage detector.

ÖZET

UZAKTAN ALGILAMA GÖRÜNTÜLERDE TEK AŞAMALI YÖNLÜ NESNE TANIMA

Atakan Serbes
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Danışmanı: Selim Aksoy
Mart 2022

Teknolojideki ilerleme, yüksek teknoloji uydulardan ve uçak algılayıcılarından yüksek miktarda bilgi birikimine sebep oldu. Söz konusu bu platformlardan elde edilen bu yüksek çözünürlüklü görüntüler insanların yeryüzünü daha iyi anlamasını sağladı. Uzaktan algılama görüntü analizi alanındaki ilk çalışmalar, görüntü sınıflandırma alanında manzara, arazi ve uçsuz bucaksız alanları anlamlandırmaya çalışan çalışmalardı. Son zamanlarda çok yüksek çözünürlüklü elektro-optik görüntülerde nesne tanıma ve görüntü bölütleme popüler hale geldi.

Modern nesne tanıyıcıların iki tür mimarisi vardır. İki aşamalı nesne tanıyıcıları, bir bölge önerme mekanizması ile olası nesne konumlarını tarar ve nesneleri sınıflandırmak için bu önceden seçilmiş bölgeler üzerinde bağlanım yapar. Öte yandan, tek aşamalı nesne tanıyıcıları, herhangi bir ön seçim olmaksızın, görüntü üzerinde genel bağlanım ve sınıflandırma gerçekleştirir. Bölge önerme mekanizması doğruluğu artırır, ancak sezim hızını yavaşlatır. Bu sebeple, tek aşamalı nesne tanıyıcılar, gerçek zamanlı sezim için daha iyi bir seçenek olmalarına rağmen, daha düşük doğruluğa sahiplerdir. Halka açık veri kümeleri genellikle yatay sınırlayıcı kutu etiketi sağlarlar. Bununla birlikte, uzaktan algılama görüntüleri, yönlü sınırlayıcı kutular için çok daha uygun olan küçük, sıkışık veya büyük en-boy oranlı nesneler ile doludur.

On beş nesne sınıfını içeren en kapsamlı uzaktan algılama nesne tanıma veri setlerinden (DOTA) birinde tek aşamalı nesne tanımaya odaklanmaktayız. Tek aşamalı nesne tanıyıcılarımız, en öne çıkan tek aşamalı nesne tanıma mimarilerinden birine (YOLO) dayanmaktadır. Yatay ve yönlü sınırlayıcı kutu etiketleri, yönlendirilmiş nesne tanıma ayarının genel olarak eksen hizalı nesne algılama ayarına göre üstün bir performansa sahip olduğunu göstermek ve ayrıca tek

aşamalı nesne tanımanın iki aşamalı nesne tanıma ile karşılaştırıldığında rekabetçi sonuçlar elde edebildiğini ispatlamak için kullanılmıştır. Yönlü nesne tanıma yaklaşımımız, açılı bağlantı problemini, bir sınıflandırma problemine dönüştürür ve tek aşamalı nesne tanıyıcılara kolayca uygulanabilir. Deneylerimiz, açısal sınıflandırmaya dayalı yönlü nesne tanıyıcısının, yatay nesne tanıyıcısından daha iyi bir performans gösterdiğini ispatlamaktadır. Ayrıca çalışmamız, doğrulama veri setinde yayınlanmış çalışmalardaki çift aşamalı nesne tanıma yöntemlerine göre %3.07, çapraz nesne tanıma yöntemlerine göre %12.84 gelişme göstermektedir.

Anahtar sözcükler: Uzaktan algılama görüntü analizi, derin öğrenme, tek aşamalı nesne tanıma, yönlü nesne tanıma.

Acknowledgement

I would like to express my sincerest gratitude to my advisor Prof. Dr. Selim Aksoy, for his great support, guidance, and vast patience throughout my master's studies. He was always ready to help and had an incredible tolerance during this challenging journey in which I gained a lot of experience and acquired invaluable skills.

I would also like to thank Assist. Prof. Dr. Shervin Rahimzadeh Arashloo and Assist. Prof. Dr. Ramazan Gökberk Cinbiş for being members of the thesis committee and their feedback.

I would like to express my deepest gratitude to my mother and father for their constant support throughout my life and my studies.

Lastly, I would like to express my special thanks to my friends Alper, Emir, and many of whom I cannot name here, who were always ready to help and be there for me with their support and love.

Contents

1	Introduction	1
2	Related Literature	5
2.1	Object Detection	5
2.1.1	Two-Stage Object Detection Architecture	6
2.1.2	One-Stage Object Detection Architecture	8
2.2	Object Detection in Remote Sensing	13
3	Methods	16
3.1	Improved YOLO Architecture	16
3.1.1	Backbone	18
3.1.2	Spatial Pyramid Pooling	18
3.1.3	Path Aggregation	19
3.1.4	YOLO Head	20
3.2	Rotated YOLO Architecture	21

3.2.1	Regression Based Rotation Detection	21
3.2.2	Boundary Problems	23
3.2.3	Angular Prediction as Classification	25
3.2.4	Circular Smooth Label	25
3.2.5	Loss Function	28
4	Experiments & Results	30
4.1	Data Set	30
4.2	Evaluation Metrics	32
4.3	Experimental Setup	34
4.4	Implementation Details	37
4.4.1	Improved YOLO Architecture	37
4.4.2	Rotated YOLO Architecture	38
4.5	Results	39
4.5.1	Improved YOLO Architecture Results	39
4.5.2	Rotated YOLO Architecture Results	41
4.5.3	Discussion	43
4.5.4	Qualitative Results	48
5	Conclusion	53

List of Figures

1.1	Illustration of the inefficiency of the axis aligned bounding boxes.	3
2.1	Illustration of a common two-stage object detection pipeline. . . .	7
2.2	Illustration of a common one-stage object detection pipeline. . . .	8
2.3	Illustration of three scale prediction on YOLOv3.	11
2.4	An illustration of input with many negative samples and only two positive samples across the image.	12
3.1	Improved YOLO Architecture.	17
3.2	Baseline DenseNet Architecture.	18
3.3	Spatial pyramid pooling.	19
3.4	FPN and PANet.	19
3.5	Concatenation in modified PAN.	20
3.6	YOLO Head Illustration.	20
3.7	Five-parameter with 90 ° range.	21

3.8	Five-parameter with 180° range.	22
3.9	Eight-parameter bounding box (blue) and ordering of corner points (green).	23
3.10	Boundary problem of five-parameter with 90° range bounding box.	24
3.11	Boundary problem of five-parameter with 180° range bounding box.	24
3.12	One-hot label encoding used on angular range.	25
3.13	Circular Smooth Label with a Gaussian Window Function.	27
3.14	Gaussian Circular Smooth Label with labels of 78 and 88.	27
4.1	Example objects and annotations from DOTA data set.	31
4.2	Intersection over union.	33
4.3	Mosaic data augmentation.	37
4.4	Confusion matrix for horizontal object detector.	41
4.5	Confusion matrix for rotated object detector.	43
4.6	Impact of region proposal size on normalized validation accuracy.	44
4.7	Samples from the validation data set that includes plane and heli- copter classes together.	45
4.8	Samples from tennis court class.	47
4.9	Fine grained class recognition problem.	48
4.10	Ground truth labels for large and small vehicles from Figure 4.9. .	49

4.11 Ship detection difference between horizontal and oriented detectors in condensed setting.	50
4.12 Bridge detection difference by horizontal and oriented detectors. .	51
4.13 Large and small vehicle detection difference by horizontal and ori- ented detectors.	52



List of Tables

4.1	Class names and number of instances in the data set	32
4.2	Training/Validation/Test split of DOTA data set.	35
4.3	Image numbers after image split.	35
4.4	Class names and number of samples in each class in the data set after train and validation split.	35
4.5	Numerical results (mAP@.5) after 140 epochs of training for hori- zontal object detection.	40
4.6	Numerical results (mAP@.5) after 140 epochs of training for ro- tated object detector for different r values in Gaussian CSL function.	42
4.7	Numerical results (4-mAP@.5) for the four large aspect ratio object classes with circular smooth label and horizontal object detectors.	46
4.8	Evaluation results on DOTA validation set.	48

Chapter 1

Introduction

Remote sensing provides information about Earth via the use of aircraft or satellite images. The advance in technology enabled humans to observe the world differently through remote sensing imagery. Some of the ways that remote sensing is put into use are environmental assessment, agriculture, renewable natural resource management, meteorology, mapping, and military surveillance and reconnaissance [1]. With the proliferation of satellite and aircraft platforms and the increasing amount of Earth observation data, remote sensing image analysis is as crucial as ever to understand and analyze our planet.

Ever-increasing sensor technology provides very high-resolution remote sensing images. The details and information in the images have changed year over year. When the images contained only low-resolution scenes [2] the problem was classifying the images. As the information in the images increased, objects in the images have become meaningful, and properly identifying and localizing objects has become a widespread problem known as object detection. With abundant remote sensing data, interpreting the objects in images has become an important pillar of remote sensing image analysis. Before deep learning models, traditional machine learning methods were used to interpret remote sensing data. Convolutional Neural Networks (CNNs) have made it possible to obtain state-of-the-art results over traditional methods in the areas of image classification [3], object

detection [4] and image segmentation [5].

Sliding window methods were being used for object detection before deep learning methods improved upon them in terms of efficiency and accuracy. The problem with traditional sliding window method is that the traditional methods are memory intensive approaches as opposed to modern one-stage [6] [7] [8] and two-stage object detectors [4] [9].

Even though object detection is a prominent topic in Computer Vision and Deep Learning, object detection in remote sensing images is not as advanced as natural scene object detection. The difference between the two topics stems from the object of interest's appearance. Objects in natural scene images appear standing and in profile whereas objects are observed from bird's-eye view in remote sensing images. The difference between the two fields also exists in the data set domain. Remote sensing image data sets [10] [11] [12] are not as inclusive as natural scene object detection data sets [13] [14] [15] [16]. A fundamental problem that leads to inadequate detection results in remote sensing object detection is overlapping bounding boxes or redundant bounding box area, which does not include parts of the object of interest as illustrated in the left part of Figure 1.1. In the case of a bridge object, the size variation can be very high, and a bounding box representing a bridge may have a meager percentage of its area that is bridge. Many small objects in remote sensing images are crowded, such as vehicles in a parking lot or ships in a harbor. Crowded small objects' bounding boxes overlap each other, which may cause the detection model to miss detect some of these small objects.

To overcome the misdetection and inadequate detection issues, rotated bounding boxes are implemented instead of axis-aligned bounding boxes, which only have four points and are aligned with x and y axes. Axis-aligned bounding boxes either include four edges of the bounding box or center point coordinates, width, and length of the box. In the rotated bounding box case, it generally has a fifth parameter that is the angle from one of the edges counter clock wise. With the use of rotated bounding boxes, the bounding box fully encloses the object of interest as illustrated in the right part of Figure 1.1. Rotated bounding box prevents

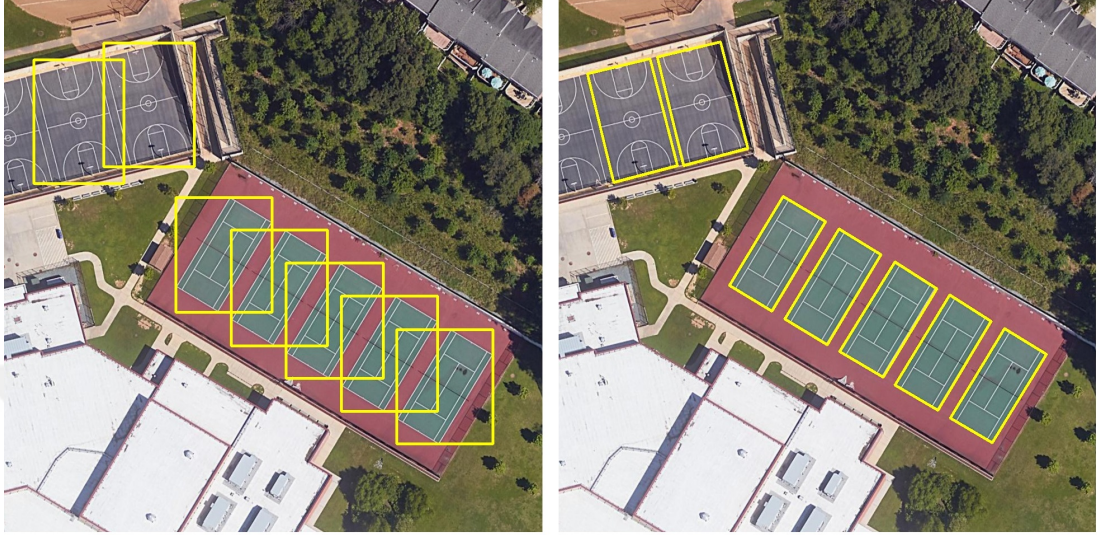


Figure 1.1: Illustration of the inefficiency of the axis aligned bounding boxes. The overlaps of bounding boxes may cause misdetection because of the elimination process of positive samples after detection results. The result with the highest IoU with ground truth will be left in the crowded area of overlapping bounding boxes.

small object box overlaps, which might cause misdetection of some objects within the cluster of objects.

In our work, we first experiment with an advanced one-stage object detector using horizontal bounding boxes and then implement the rotated bounding box to detect objects in the oriented setting. Even though two-stage object detectors have generally higher accuracy, we get competitive results with improved YOLO architecture. The contributions of the thesis are the implementation of rotated bounding box on the improved YOLO architecture with a method of classification for the angle instead of regression which allows the detector to detect objects of interest in an oriented manner. We experiment and analyze the implementation with different parameters to observe the effects on the detections for the remote sensing images. In experiments, we showed that for the validation set results, angular classification approach with improved YOLO architecture can improve detections over recent two-stage and anchor-free detection methods. We also show that the angular classification approach is best suited for large aspect ratio objects with boundary issues during regression scenarios.

This thesis is organized as follows. In Chapter 2, background research on object detection and remote sensing object detection is discussed. In Chapter 3, methods that were implemented are discussed. In Chapter 4, the characteristics of the data set that was used during the studies, evaluation metrics, experiments and results are discussed. Finally, the conclusion is presented in Chapter 5.



Chapter 2

Related Literature

This chapter presents related work on the research done on this thesis. First, related work on object detection is presented in two-stage and one-stage object detection groups. Then, related work on remote sensing object detection is discussed. Lastly, related work on rotated object detection is presented.

2.1 Object Detection

Image classification problems assume there is a single object or scene in the image and there is no spatial extent. The aim of object detection, on the other hand, requires properly identifying and localizing objects within the image using bounding boxes that surround each instance. An image can include multiple instances of objects of the same or different classes. Modern object detectors have two types of architecture which are single-stage and two-stage. Two-stage object detectors such as [4] [9] use a region proposal stage to generate a sparse set of candidate locations of objects and classify and regress their locations in their second stage whereas one-stage object detectors such as [6] [7] [8] regress candidate locations of objects. Two-stage detectors have higher accuracy rates with slow speeds, and one-stage detectors have lower accuracy rates with faster

speeds compared with their counterparts.

2.1.1 Two-Stage Object Detection Architecture

Two-stage or proposal-based object detectors generate a set of candidate object locations or region of interests (RoIs) in the first stage and classify and localize objects of interest in RoIs in the second stage of the architecture.

The modern two-stage detectors are all built on R-CNN family [4] [9] [17]. Girshick *et al.* [17] introduced R-CNN in 2014, which became the building block of two-stage object detectors. It uses a proposal mechanism such as selective search [18] to obtain candidate region proposals which are resized into a fixed size to be fed into a convolutional neural network (CNN). The CNN outputs predicted class scores and bounding box refinements. Based on the scores, the higher scored region proposals are selected and compared with ground truth labels. The comparison is made with IoU (intersection over union) metric to select the region proposals with high scores that overlap a certain amount with the ground truth. The certain amount is called the threshold and is usually bigger than 0.50. After the elimination of region proposals with low IoU, a technique to eliminate redundant predictions is used to make sure that one object of interest is detected by the most accurate prediction. The elimination of redundant detections is done by non-maximum suppression(NMS).

The problem with R-CNN was that it is very memory extensive. Fast R-CNN [4] was introduced to mitigate the problem of running every region proposal through CNN. Instead of individual runs, the whole image is run through a CNN to obtain a feature map of the whole image. This way, the main computation happens in the backbone network. Then a region proposal method is used on the feature map to obtain the region of interests (RoIs) using RoI Pooling [4] method. RoI pooling results in cropped and resized features which are then fed into CNNs for classification and bounding box refinement.

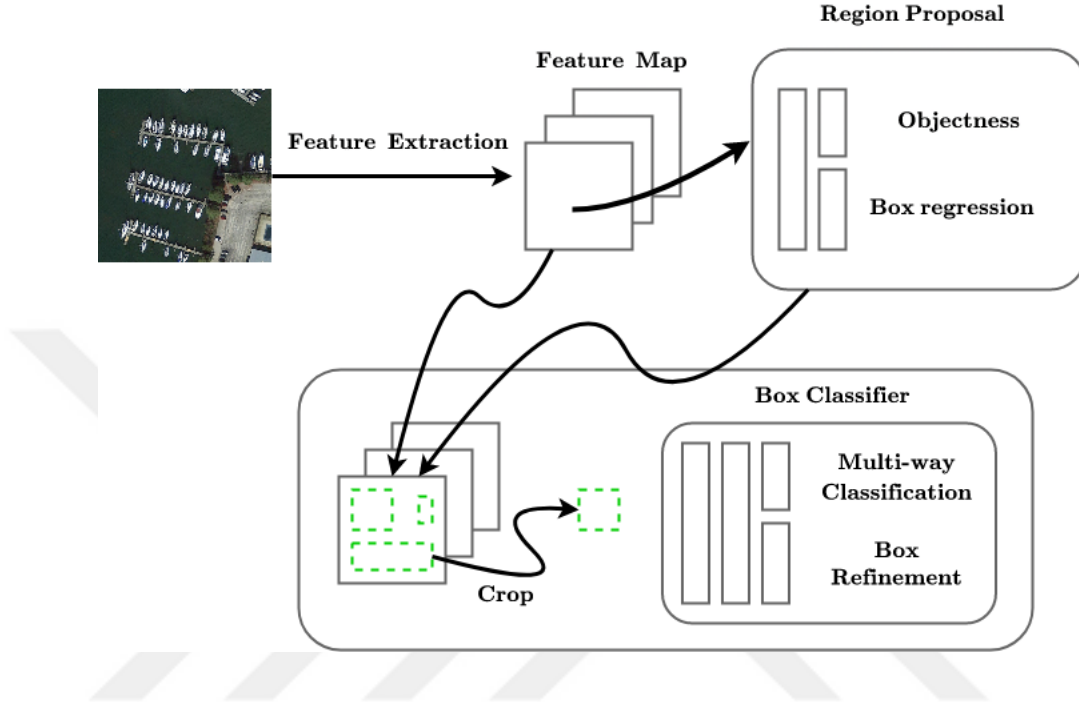


Figure 2.1: Illustration of a common two-stage object detection pipeline. It consists of a feature extractor, region proposal network and a box classification and regression part.

The final member of the R-CNN family is Faster R-CNN [9] in which Ren *et al.* add a region proposal network (RPN) after the feature extraction (backbone) to predict the region proposals. The image is put through the backbone network for feature extraction to obtain the feature map. Image level features are then passed to RPN to obtain region proposals. RPN uses predefined bounding boxes laid on the image with different scale and aspect ratios which are also called anchor boxes, to determine the probability of an anchor having an object of interest (objectness). RPN also uses box regression on anchors to refine the coordinates of the proposals. Then by using the image feature map and the outputs of RPN, RoI pooling crops the region proposals to be used for final classification and bounding box regression. Excluding the RPN part, Faster R-CNN and Fast R-CNN architectures are the same. A classic two-stage object detection pipeline that is based on Faster R-CNN is illustrated in Figure 2.1.

2.1.2 One-Stage Object Detection Architecture

Overfeat [19] was one of the first single-stage object detectors. Sermanet *et al.* [19] used a traditional sliding window approach to outperform two-stage object detectors that used a region proposal mechanism. After [19], the pioneers of single-stage object detectors [7] and [6] became popular. These architectures perform a global classification and regression over image pixels to classify and detect objects instead of performing these tasks from region proposal locations.

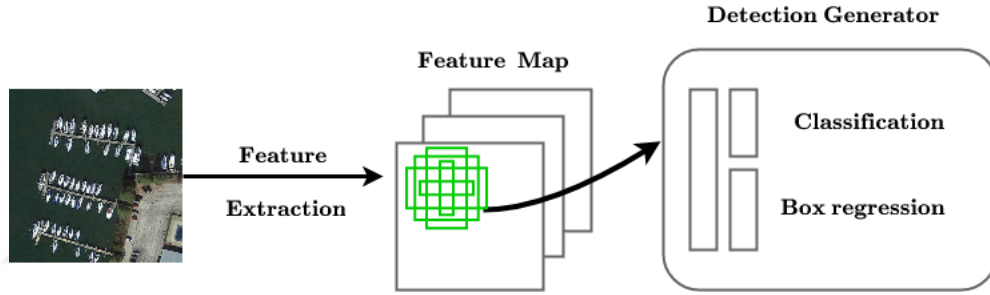


Figure 2.2: Illustration of a common one-stage object detection pipeline. It consists of a feature extractor and a box classification and regression part.

One of the essential parts of single-stage object detectors is having a collection of predefined boxes with various scales overlaid on the image at different locations and aspect ratios. These are called anchors or priors, and they have ideal shapes and sizes of possible objects. Different spatial locations make it possible to have an anchor overlap with one object in the image. By having different shapes, it is possible to have an anchor overlap with differently shaped objects in the image. By having different sizes, it is possible to have an anchor overlap with small, medium, or large-sized objects within the image. Instead of region proposals that are likely to include objects, one-stage object detectors have a large set of candidate object locations with different shapes and sizes arranged in order across an image.

It is because of the anchors that one-stage detectors have high efficiency but less accuracy. It is possible to increase the accuracy of the models since the

problem of object detection becomes an image classification problem. However, only a small amount of anchors overlap with ground truth boxes meaning there are excessive negative samples and a relatively smaller number of positive samples. This results in the foreground and background class imbalance problems and decreases the accuracy of models. An illustration of a common one-stage object detection pipeline is provided in Figure 2.2.

2.1.2.1 YOLO Family of Architectures

YOLO architecture is one of the pioneers of single-stage object detectors. It is mainly used for real-time object detection because of its high speed.

In the first YOLO architecture, Redmon *et al.* uses a simple convolutional neural network to extract bounding boxes directly from the image [6]. Compared to its alternative two-stage object detector [17] which uses a region proposal stage before detecting objects within the object proposals, YOLO is an end-to-end unified network to perform the tasks of a two-stage object detector as a simple network. The image is divided into grids and within grids lies bounding boxes predicted with their confidence scores. The bounding box contains five parameters regarding the object, which are $(x, y, w, h, confidence)$. Center coordinates (x, y) and width and height (w, h) are relative to the boundaries of grids and relative to the image, respectively. The values of (x, y, w, h) are normalized to be between 0 and 1. The confidence score is the probability of an object being in that particular bounding box and is measured by the IoU metric between the bounding box and a ground truth. Each grid cell within the image also has a probability that is conditional on whether an object exists in the cell or not. The resulting speed of the unified network is a thousand times faster than [17] and a hundred times faster than [4]. In the loss function, YOLO uses the sum of squared errors (SSE) for x, y, w , and h values.

Grid design in [6] forces each grid cell to have only one object assigned to it. This is why YOLO performs poorly in the case of small object detection since the probability of small or cluttered objects lying in the same grid cell is high.

After YOLO, Redmon *et al.* [20] proposed another network named YOLO9000 that has many improvements over shortcomings of the first design. Firstly the Batch Normalization [21] is used on every layer, which helps regularize the model. Secondly, Redmon *et al.* fine-tune the classifier on ImageNet [22] at full resolution to increase the accuracy of the network. The third and the most important improvement in YOLO9000 is using anchor boxes to predict bounding boxes. However, the anchor boxes directly have hand-picked aspect ratios, and these ratios might not be suitable for the objects in the data set, which is why Redmon *et al.* use k-means clustering with the distance metric,

$$d(\text{box}, \text{centroid}) = 1 - \text{IOU}(\text{box}, \text{centroid})$$

instead of k-means with euclidean distance to find dataset suitable anchor boxes. In addition to these improvements, the prediction of center locations is limited in between 0 and 1 to increase the stability of grid cell predictions, a passthrough layer is added to concatenate high resolution features with low resolution features to enable fine-grained feature detection, and lastly, random changes to input image size are made for multi-scale training which ensures that the network can predict with different input image sizes.

In the latest version of YOLO, which is named YOLOv3 [23], Redmon *et al.* uses a better and a much deeper feature extractor named DarkNet-53 [23] which has shortcut connections and is bigger in network size. Feature Pyramid Network (FPN) [24] like structure is used for multi-scale feature extraction that helps YOLOv3 to predict bounding boxes with three different sizes. YOLOv3 design uses the previous layer’s information by upsampling the previous feature maps and concatenating the features with another earlier feature map to increase the understandability of semantic information and is illustrated in Figure 2.3. This FPN-like structure helps YOLOv3 achieve higher average precision results for small objects.

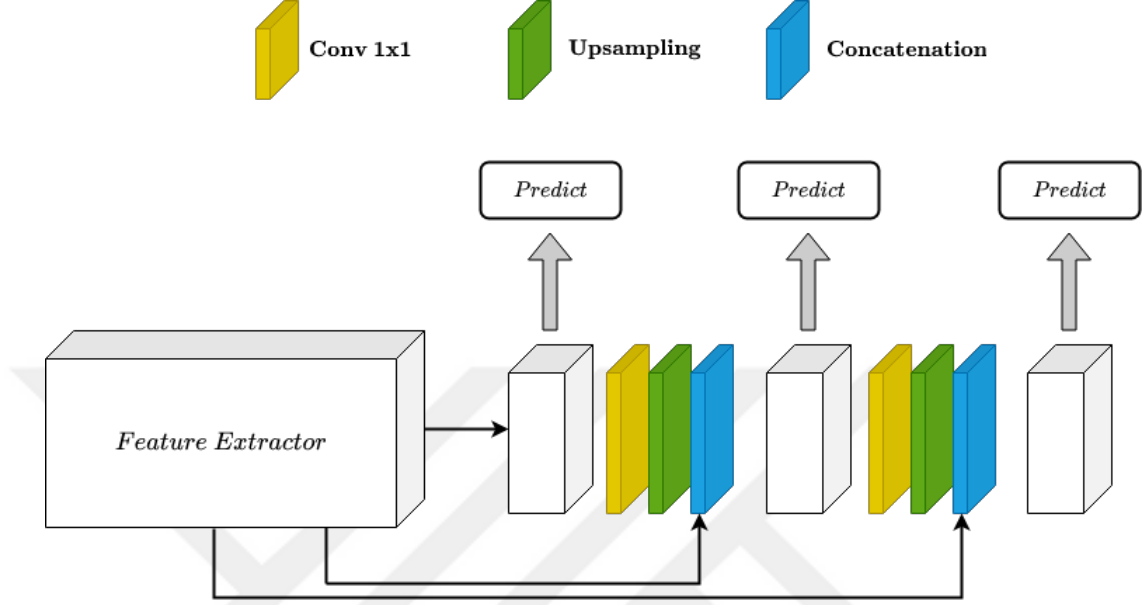


Figure 2.3: Illustration of three scale prediction on YOLOv3. 1x1 convolutional layer, upsampling and concatenation operation are used. Later concatenation uses earlier layer information. Structure resembles the feature pyramid network mechanism.

2.1.2.2 RetinaNet

To solve the class imbalance problem that stems from anchors, Lin *et al.* [25] propose a new loss. Focal loss in RetinaNet is a 'soft' hard-example mining to mitigate the data imbalance problem.

RetinaNet has a ResNet [26] feature extractor network with a feature pyramid net which assembles the FPN [24] structure which is followed by two subnets for classification and regression. RetinaNet uses anchor boxes like YOLO family [6] [20] [23] and SSD [7]. Even though the number of boxes in the design is substantially higher than other one-stage detectors, RetinaNet uses a unique loss to mitigate the issue of extreme class imbalance. The class imbalance problem is illustrated in Figure 2.4 which shows the dominance of easy examples in an object detection input image. Focal loss [25] down-weights the easy examples,

and thus the training procedure can focus on the hard negative samples. Focal loss is defined as,

$$\text{FL}(p_t) = -(1 - p_t)^\gamma \log(p_t)$$

A factor of $(1 - p_t)^\gamma$ is added to cross entropy (CE) loss and γ is used as the parameter of focus to down-weight the easy samples. It can be seen that when γ is 0, focal loss becomes the cross entropy loss. Experiments done by Lin *et al.* in [25] show that best result is obtained with $\gamma = 2$. In the experiments, Lin *et al.* observe that focal loss succeeds in making training focus on hard negative examples, which makes RetinaNet a network with the speed of a one-stage detector and an accuracy of a two-stage detector.

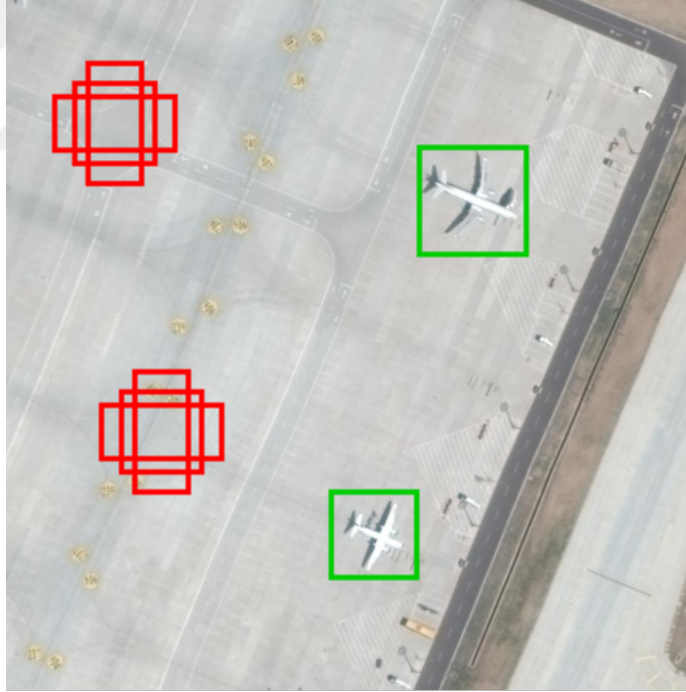


Figure 2.4: An illustration of input with many negative samples and only two positive samples across the image. One-stage detectors face such scenarios which dominate the training procedure with background examples.

2.2 Object Detection in Remote Sensing

One of the first high resolution object detection datasets in remote sensing is NWPU VHR-10 [10] [27]. The dataset contains 800 very high resolution images obtained from Google Earth and Vaihingen data [28]. Cheng *et al.* in [27] use traditional machine learning methods to obtain detection results.

Zhu *et al.* [29] study airplane and vehicle detection in UCAS-AOD data set. Zhu *et al.* in [29] make use of deep convolutional neural networks in feature extraction part.

DOTA data set [11] is one of relatively recent large-scale remote sensing object detection data set that includes 15 common object classes. Xia *et al.* provide [11] some baseline results for the two annotation formats, which are horizontal and oriented bounding box labels. The detection models used for baseline results are SSD [7], Faster R-CNN [9] and YOLOv2 [20]. Faster R-CNN achieves the highest accuracy among baseline results. For the oriented bounding box format, inference by Faster R-CNN trained on both oriented and horizontal bounding box formats was used. According to [11], using the horizontal bounding boxes for training, one-stage object detectors get between 29 to 39 mAP, and two-stage object detectors get up to 60 mAP. Training from oriented bounding boxes results in one stage object detectors having accuracies between 17 to 25 mAP and two-stage object detectors having accuracies between 39 to 54 mAP.

According to [30] both one-stage and two-stage object detection methods have some common problems. One of the problems is that existing bounding box regression loss has no direct correlation with the metric it is trying to optimize, the IoU. They propose improved generalized IoU (IGIoU) to solve this problem. The other problem is with two-stage methods and region proposal mechanisms. Each proposal maps into a single feature map which that does not use multi-level features. Feature extraction of region proposals does not put multi-level features into use, and this information is only used in the RPN stage of two-stage detectors. Qian *et al.* [30] propose a Multi-level features fusion (MLFF) module

that makes full use of multi-level features.

Rotation detection is applicable to scene text detection, face detection, and remote sensing object detection scenarios. The objects in remote sensing images are naturally positioned arbitrarily unlike their horizontal positions in generic object detection data sets like VOC2007 [14], VOC2012 [15] or COCO [13]. Consequently, small and cluttered objects arbitrarily positioned in the remote sensing images face lower detection rates or false negatives.

The first rotated object detectors were designed for specific categories of objects such as vehicles, ships, and planes. Tang *et al.* uses the fifth parameter of angle together with the bounding box parameters to be able to detect the orientation of vehicles using a design based on SSD [7] named Oriented SSD [31]. Liu *et al.* [32] uses a rotated region of interest pooling layer, a new rotated bounding box regression model, and a new non-maximum suppression method to detect ships in a rotated manner. Zhang *et al.* proposed a new arbitrary oriented ship detection network [33] with rotated RPN to generate rotated region proposals with a model that employs rotated region of interest pooling operation to extract features. Liu *et al.* uses prior rotated boxes with many different angles for predictions [34] which narrows down the detection task to sub-tasks with different angle categories. After large-scale remote sensing data sets such as [11] became public, the rotated object detection task also expanded to multi categories.

Azimi *et al.* use Image Cascade Network (ICN) [35] and Feature Pyramid Network (FPN) [24] to extract rich multi-scale features, which are then fed into rotation region proposal networks and multi-scale rotated RoI network to be able to detect arbitrary oriented objects on DOTA data set [11]. The network outputs results in a quadrilateral format, which is different than OpenCV [36] and long-edge definitions of oriented bounding boxes.

Ding *et al.* study RoI transformation [37] to apply spatial transformations on RoIs using oriented bounding boxes. The RoI transformer is a tool to embed into detectors to make oriented bounding box detection possible by geometry transformation. Authors transform the horizontal region of interests into a rotated

regions of interest which are then used to extract rotation-invariant features for prediction. Ding *et al.* also report their results on their public DOTA data set which shows improved results over the baseline results in [11].

Yang *et al.* work on top of RetinaNet [25] to design refined rotation RetinaNet that is composed of a single-stage detector with a feature refinement module [38] to solve the feature misalignment problem. Yang *et al.* also devise an approximate SkewIoU loss for a better rotation detection.

Many oriented object detectors suffer from boundary discontinuity which is caused by angular periodicity. This problem is the result of the optimal prediction residing outside the defined range of bounding box angle parameter, which is the range of $[-90, 0)$ for OpenCV [36] five-parameter annotation and the range of $[-90, 90)$ for the long edge five-parameter annotation.

Yang *et al.* propose a new loss function called IoU-smooth L1 loss [39] to factor in the IoU as a constant in loss to improve the smooth L1 loss. IoU-smooth L1 loss mitigates the effects of boundary problems in rotated object detection and the model. Qian *et al.* propose a new loss called modulated rotation loss [40] to address the issue of sudden loss increase at the boundaries in the case of boundary scenarios. IoU-smooth L1 loss [39] and modulated rotation loss [40] are designed for five-parameter and eight-parameter rotated bounding box regression respectively.

The boundary problem which happens when a regression is outside the defined range of the rotated bounding box can be mitigated by transforming the angular regression to an angular classification problem.

Yang *et al.* propose a new technique named circular smooth label (CSL) [41] to transform angular regression to a classification problem that can be embedded into regression-based methods. A circular smooth label provides a way to be able to detect angles with high precision and fine granularity contrary to older and coarse granularity classification methods.

Chapter 3

Methods

In this chapter, we describe the methods used during the experiments. First, the horizontal object detector architecture is explained, and then the rotated object detection method is presented.

3.1 Improved YOLO Architecture

The improved YOLO architecture is based on one of the first one-stage object detectors [6]. It has a backbone, a neck, and a head of the network. The neck in the structure is usually used to interact with feature maps from different layers. The backbone of the network is CSPDarknet53 [42]. It is a backbone that uses CSPNet [43] strategy and DarkNet-53 [23]. In addition, spatial pyramid pooling [44] is used over the backbone to increase the receptive field. The neck of the network is PANet [45] path-aggregation network to enhance information propagation by interacting with feature maps on various layers. Finally, the head of the network is based on YOLOv3 [23]. The architecture of the improved YOLO network is illustrated in Figure 3.1.

3.1.1 Backbone

Backbone of improved YOLO architecture is based on [46] and Cross-Stage-Partial-Networks (CSP) [43]. DenseNet is a backbone architecture that is formed by Dense and Transition Blocks as illustrated in Figure 3.2. Just like in ResNet [26] the idea of the addition of information from previous layers into the output of the present layer is used.

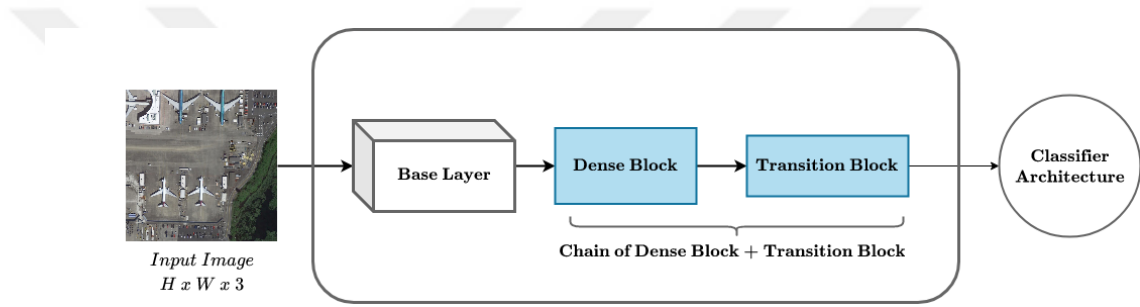


Figure 3.2: Baseline DenseNet architecture that is formed by chains of Dense and Transition Blocks.

However, the problem with DenseNet architecture is replicated gradients. The input of the k^{th} dense block will be a concatenation of previous $k - 1$ dense layers. This causes inefficient optimizations during gradient descent. To mitigate the problem of replicated gradients Cross Stage Partial Network strategy is used. By breaking the base layer into two parts, the Dense Block calculations decrease by half.

3.1.2 Spatial Pyramid Pooling

Spatial pyramid pooling (SPP) [44] helps bypass the requirement of fixed size inputs into convolutional neural networks (CNNs). It generates fixed-length outputs without the need to distort images that are fed into CNNs, and it is illustrated in Figure 3.3. However, in their design of YOLOv3 [23] Redmon *et al.* modify the SPP module and increase the receptive field of the backbone feature.

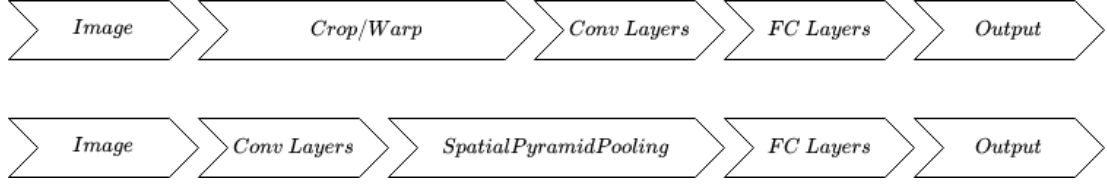


Figure 3.3: Spatial pyramid pooling operation removes the need of distorting the input image.

3.1.3 Path Aggregation

The neck part of the object detectors is used to make semantic extraction from the inputs richer by fusing multi-scale features from different levels. Path aggregation network boosts the feature pyramid [24] by adding a bottom-up path augmentation onto FPN. The FPN structure is illustrated in the left part of Figure 3.4 with a top-down pathway. PANet structure has an extra pathway to fuse features, as illustrated in the right part of Figure 3.4.

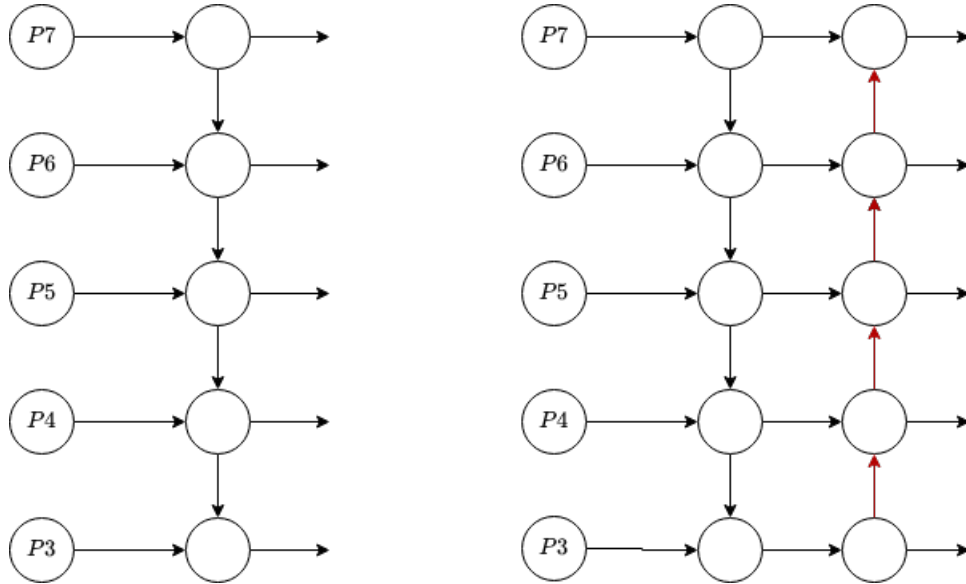


Figure 3.4: FPN with top-down pathway on the left and PANet with additional bottom-up pathway on the right for multi-scale feature fusion.

As shown on the right side of Figure 3.5, improved YOLO architecture PAN is modified to concatenate feature maps in the bottom-up aggregation layer instead of adding them to improve the detection accuracy.

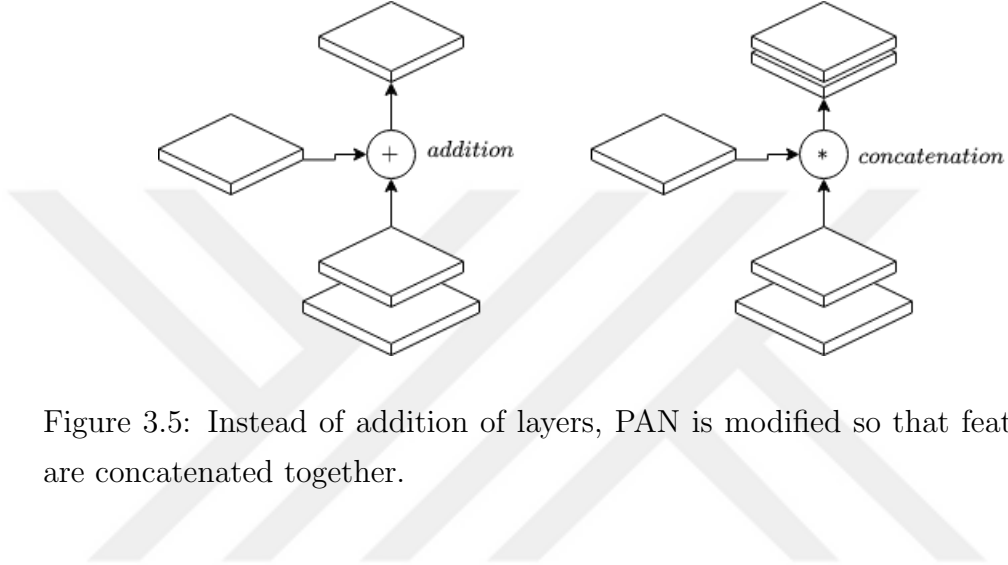


Figure 3.5: Instead of addition of layers, PAN is modified so that feature maps are concatenated together.

3.1.4 YOLO Head

The head, in other words, the dense prediction part of the detector, is based on YOLOv3 [23]. YOLO head provides three levels of detection granularity as illustrated in Figure 3.6. The different-sized features are then fused to be able to detect objects of different sizes. In the final output vector, it has the information on the predicted bounding box, including the center coordinate, height and width values, confidence score, and label of the object.

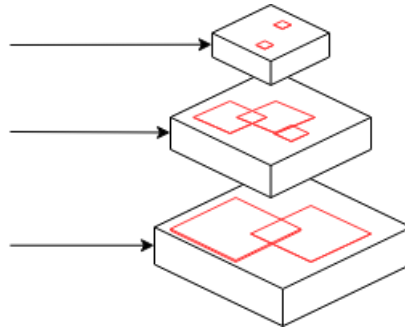


Figure 3.6: Detection in three levels of granularity with YOLO head.

3.2 Rotated YOLO Architecture

The rotated object detector is also based on the improved YOLO architecture that is explained in Section 3.1. However, in the bounding box regression part of the detector head, there are methods used to be able to regress the bounding box in an oriented manner.

3.2.1 Regression Based Rotation Detection

There are three different popular methods for oriented object detection, which have different types of definitions for bounding boxes and use different techniques for bounding box regression.

3.2.1.1 Five-parameter Regression with 90° Angular Range

The first and most used method for oriented detection is by adding a fifth parameter into (x, y, w, h) to make it (x, y, w, h, θ) . The additional parameter θ is the angle between width and x-axis as shown in Figure 3.7.

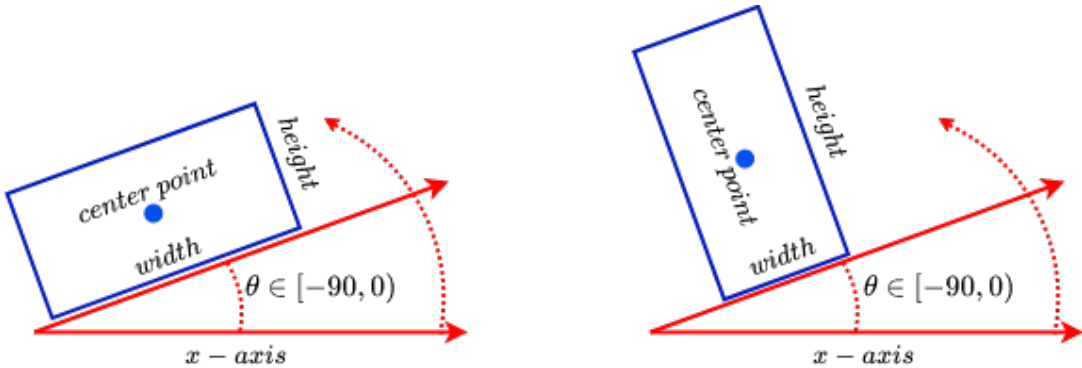


Figure 3.7: Five-parameter with 90 ° range. Influenced from [41].

3.2.1.2 Five-parameter Regression with 180° Angular Range

The second way to define a rotated bounding box is via a fifth parameter. However, in this case, the angular range is 180°, and the angle is defined by the angle between the long side (height) and the x-axis as shown in Figure 3.8.

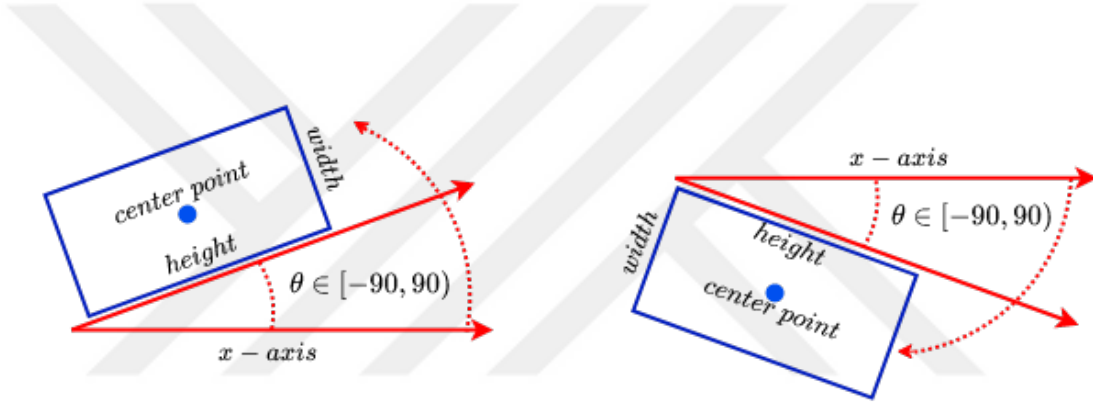


Figure 3.8: Five-parameter with 180 ° range. Influenced from [41].

3.2.1.3 Eight-parameter Regression

Eight-parameter methods try to regress the corners (a, b, c, d) or in other words points $(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$ of quadrilateral as shown in Figure 3.9. The prediction becomes a quadrilateral like the blue box in Figure 3.9. There is no parameter undefined in the coordinate system which makes it a consistent representation compared to representations with an angle.

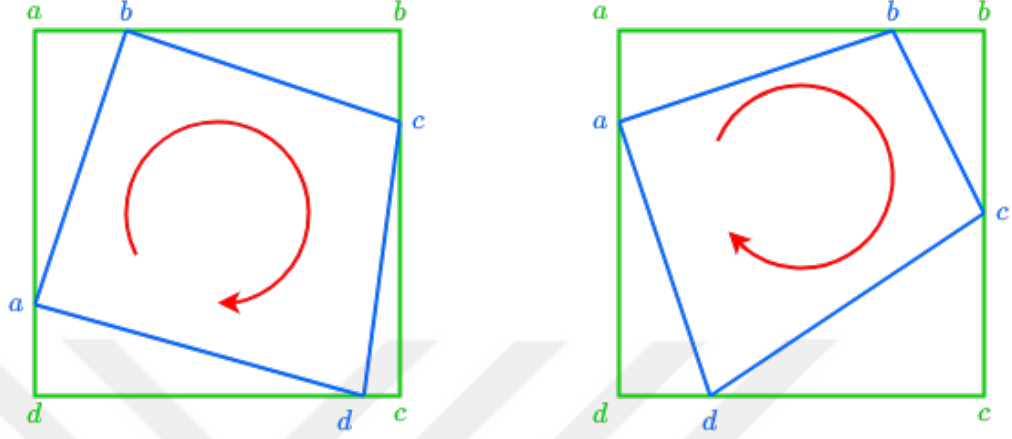


Figure 3.9: Eight-parameter bounding box in blue and ordering in green.

3.2.2 Boundary Problems

The boundary condition is defined as the angle parameter of a rotated bounding box being close to the endpoints of the angular range that defines the rotated bounding box.

In boundary cases for a five-parameter with 90° range bounding box, the ideal case for regression might be a slight rotation. However, if the slight rotation is an angular degree that is not defined within the range of 0° to 90° , the model regresses the bounding box in a nontrivial way. The ideal and actual regression for such a scenario is shown in Figure 3.10. The regression of anchor/proposal into a predicted box in the case of actual regression is by scaling the width and height. The ideal regression has a significant loss that includes the change of edges and angular degree beyond the defined range. These problems are called periodicity of angular (PoA) and exchangeability of edges (EoE) respectively in [41]. The drawings in Figure 3.10 and in Figure 3.11 are inspired from [41] where a series of different bounding box definitions were illustrated.

By the definition of the five-parameter bounding box, the regression result and ground truth box in the ideal regression case have the total opposite edges, and

the angular degree difference between the two is 90° .

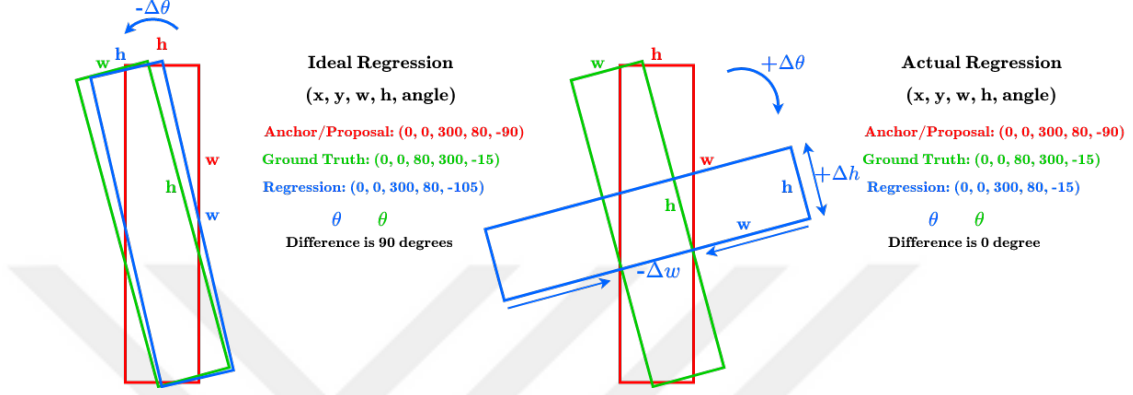


Figure 3.10: Boundary problem of five-parameter with 90° range bounding box. Influenced from [41].

Five-parameter bounding box has another definition which is defined by the angle between the long edge and x-axis, and in this case, the angular range is 180° . Even though the ideal regression in some boundary cases is just a slight change in angle, when the angle is outside the defined range of 90° to -90° , the loss is high because of periodicity of angular (PoA), which is illustrated in Figure 3.11.

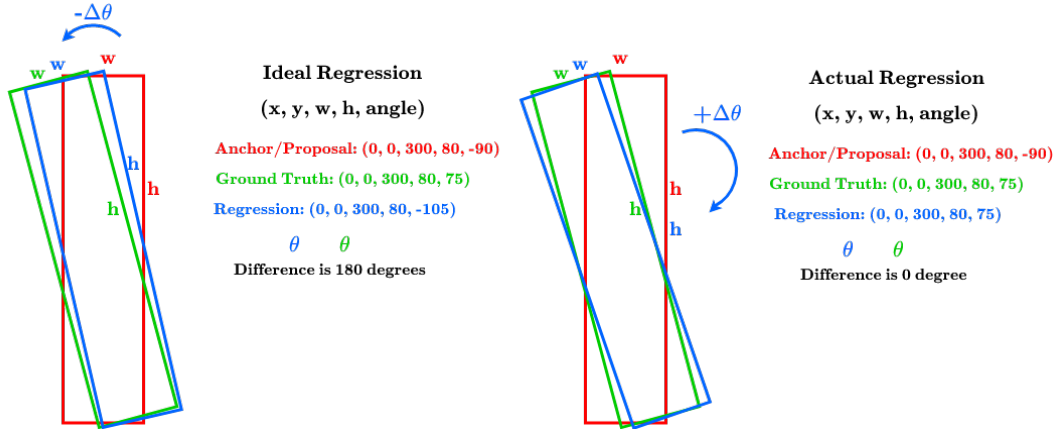


Figure 3.11: Boundary problem of five-parameter with 180° range bounding box. Influenced from [41].

There are some approaches to mitigate the issue of boundary problems in rotated bounding box regression methods. Yang *et al.* [39] introduce a new loss called IoU-Smooth L1 loss to consider IoU in the loss function. Qian *et al.* [40] also introduces a new loss named modular rotation loss to mitigate the sharp increases in losses caused by boundary cases.

3.2.3 Angular Prediction as Classification

The defined angular range of bounding boxes causes the boundary problems since the optimal regression lies outside the defined range. To mitigate this problem, angle prediction can be turned into a classification problem that introduces limits for prediction results. A vanilla solution for angular classification uses one-hot label encoding, where object angle becomes the object category. One-hot label encoding used on the angular range is shown in Figure 3.12.

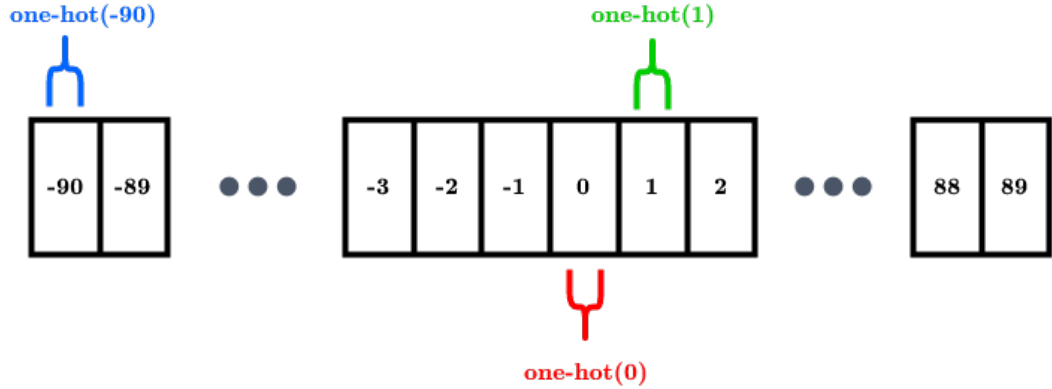


Figure 3.12: One-hot label encoding used on angular range.

3.2.4 Circular Smooth Label

The problem with one-hot label encoding that is explained in Section 3.2.3 is that it is angle agnostic. Considering Figure 3.12, a loss calculation between the ground truth $one-hot(0)$ and first prediction $one-hot(1)$ and a loss calculation

between the ground truth $one-hot(0)$ and second prediction $one-hot(-90)$ yields the same result. However, it should not be the case for rotated object detection, and the loss calculation for the prediction that is close to the ground truth should result in a smaller loss.

To overcome the issue of angle agnostic solution caused by one-hot label encoding and the boundary issues that arise mostly on high aspect ratio objects at the boundary cases, we use circular smooth label [41] to smoothly encode the label with periodicity. The circular smooth label equation is as follows:

$$CSL(x) = \begin{cases} g(x), & \theta - r < x < \theta + r \\ 0, & \text{otherwise} \end{cases} \quad (3.1)$$

where $g(x)$ is the window function. θ is the angle value from the bounding box parameters and r is the radius that extends or narrows the window function. Some window function examples can be Gaussian and pulse functions and a Gaussian window function with the angle value of 88 is illustrated in Figure 3.13.

The Gaussian function is a function of the form,

$$f(x) = a \exp\left(-\frac{(x-b)^2}{2c^2}\right) \quad (3.2)$$

where a is the height of the peak, b is the position of the function and c is the standard deviation of the function. In our case, the parameters a , b and c are 1.0, 0 and r value of the Equation 3.1 respectively. x value in Equation 3.2 is the label array which we get after we decide on the angular granularity of the solution. In our case, the five-parameter method with long-edge definition and 180° angular range is used for rotated bounding box.

When we apply the parameters to the Gaussian circular smooth label in Equation 3.2 for label values of 88 and 78, we get the function values illustrated in Figure 3.14. We can observe that Gaussian CSL provides a smooth transition between labels, and even if the labels are in the boundary conditions, the loss between a label of -88 and a label of 88 could be calculated and give information about the proximity of two labels as opposed to one-hot label encoding scenario.

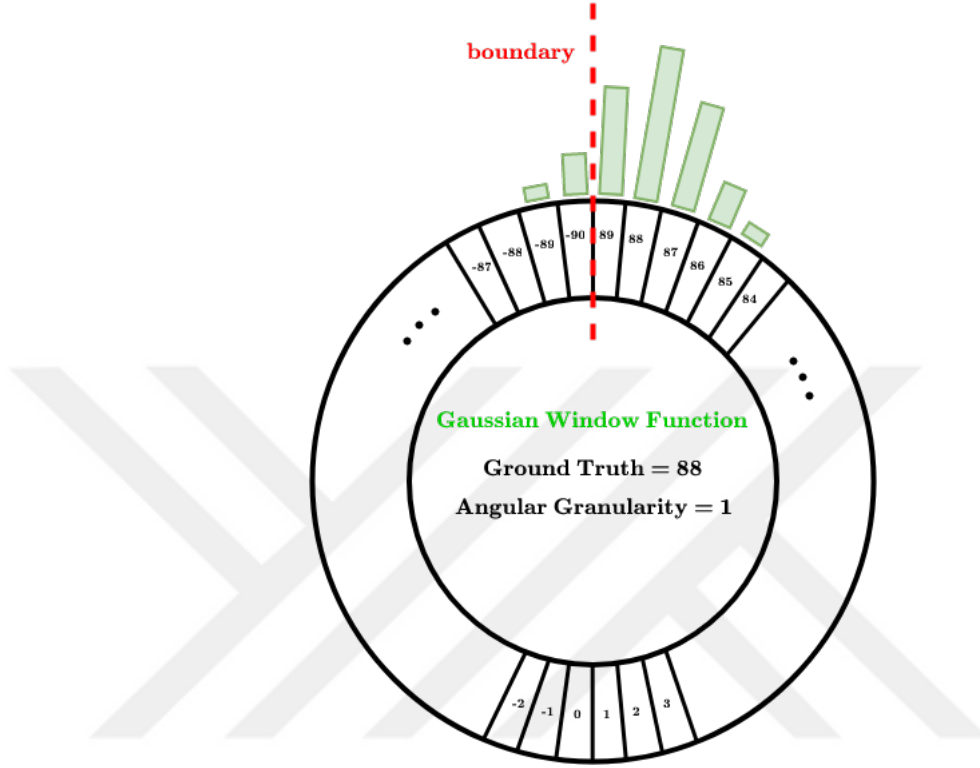


Figure 3.13: Circular Smooth Label with a Gaussian Window Function for a ground truth value of 88. Influenced from [47].

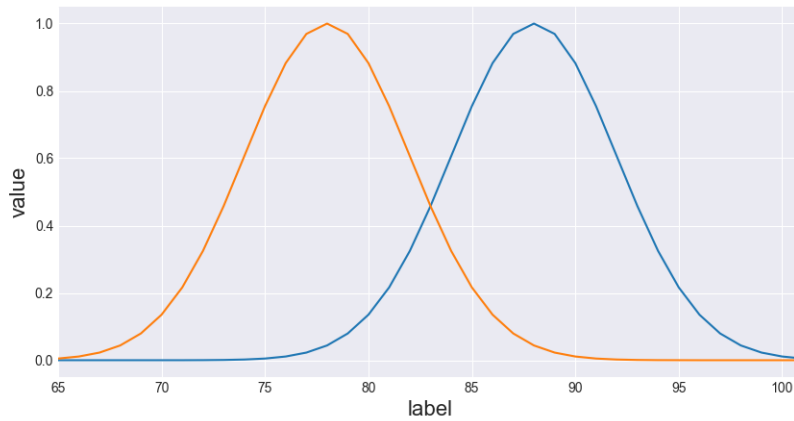


Figure 3.14: Gaussian Circular Smooth Label with labels of 78 and 88.

It can be observed from Figure 3.13 that the label that we obtain by using a

Gaussian window function and the circular smooth label has a continuous value at the boundary, which solves the boundary problem that arises in oriented object detection. It should be noted that Equation 3.1 with a small enough radius r and pulse function $g(x)$ results in one-hot label encoding that is illustrated in Figure 3.12.

3.2.5 Loss Function

Rotated YOLO architecture uses five parameters (x, y, h, w, θ) for rotated bounding box definition. The angular range is $180^\circ([-90, 90))$, and the angle is defined by the angle between the long edge (height) and the x-axis. Using the circular smooth label (CSL), there is one more angular classification layer that is optimized by angular classification loss. The multi-task loss function is defined as:

$$L_{total} = L_{reg} + L_{obj} + L_{cls} + L_{theta} \quad (3.3)$$

where L_{reg} , L_{obj} , L_{cls} , L_{theta} are regression, objectness, classification and angular classification loss respectively.

Regression loss L_{reg} uses complete IoU (CIoU) [48] which considers the central point distance and aspect ratio in addition to normal IoU metric. Objectness, classification, and angular classification metrics are all optimized by binary cross entropy (BCE) loss.

Let $\mathbf{b}$ be the center point of the predicted bounding box and let $\mathbf{b}^{gt}$ be the center point of the ground truth bounding box. Regression loss complete IoU (CIoU) can be expressed as,

$$L_{reg} = 1 - IoU + \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2} + \alpha v \quad (3.4)$$

where the smallest enclosing box encompassing the two boxes has a diagonal length of c . Euclidean distance is defined as $\rho(\cdot)$ and α is defined as

$$\alpha = \frac{v}{(1 - IoU) + v}$$

where v is the aspect ratio consistency and defined as follows

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2.$$

The objectness (confidence) loss is binary cross entropy (BCE) loss and can be expressed as,

$$\begin{aligned} L_{obj} = & -\lambda_{obj} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{obj} \left[\hat{C}_i \log(C_i) + (1 - \hat{C}_i) \log(1 - C_i) \right] \\ & -\lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{noobj} \left[\hat{C}_i \log(C_i) + (1 - \hat{C}_i) \log(1 - C_i) \right] \end{aligned} \quad (3.5)$$

where S^2 is the total number of grids, B is the number of anchors residing in each grid, λ are loss coefficients, I_{ij}^{obj} term is 1 when an object resides in the current anchor box, C_i is predicted confidence score of the current anchor box, and $\hat{C}_i$ is the ground truth confidence of the current anchor box.

The classification loss is binary cross entropy (BCE) loss and can be expressed as,

$$L_{cls} = -\lambda_{cls} \sum_{i=0}^{S^2} I_{ij}^{obj} \sum_{c \in \text{classes}} [\hat{p}_i(c) \log(p_i(c)) + (1 - \hat{p}_i(c)) \log(1 - p_i(c))] \quad (3.6)$$

where $P'_i(c)$ is the probability of detecting the object with label c , $P_i(c)$ is the probability of the object having ground truth label c and N is the number of anchors (images are divided into grids and each grid has a certain number of anchors in YOLO architecture as explained in Section 2.1.2).

The angular classification loss L_{theta} is calculated with binary cross entropy (BCE) loss and defined as,

$$L_{theta} = -\lambda_{obj} \sum_{i=0}^{S^2} \sum_{j=0}^B L_{CSL}(\theta'_n, \theta_n) \quad (3.7)$$

where L_{CSL} is binary cross entropy (BCE) loss, S^2 is total number grids, and B is the number of anchors in each grid. It also has a λ coefficient like other losses which can be configured to provide weights for losses. The loss for L_{CSL} can also be focal loss [25] which is discussed in [41].

Chapter 4

Experiments & Results

In this chapter, we describe the data set used during the experiments and then give information about the evaluation metrics. Next, we discuss the experimental setup and implementation details for the horizontal and oriented object detectors. Lastly, we present and discuss the results.

4.1 Data Set

The experiments were conducted on one of the largest object detection data sets [11] in the field of remote sensing imagery. DOTA data set contains 2806 high-resolution images and 188,282 instances with 15 common categories including plane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor bridge, large vehicle, small vehicle, helicopter, roundabout, soccer ball field and swimming pool. The image width in the data set varies between 800 and 4000 pixels. Images in the data set are collected from different platforms with multiple resolutions.

The annotations for the DOTA data set are provided in both axis-aligned and oriented (quadrilateral) format so that both axis-aligned and oriented object detection methods can be studied. The evaluation task on the data set is in two

tasks where one task is detection with horizontal (axis-aligned) bounding boxes, and the other task is detection with oriented bounding boxes. DOTA data set is challenging because of its significant number of instances. There are dense scenes that include small objects like vehicles in a parking lot which creates more challenging scenarios for detection.

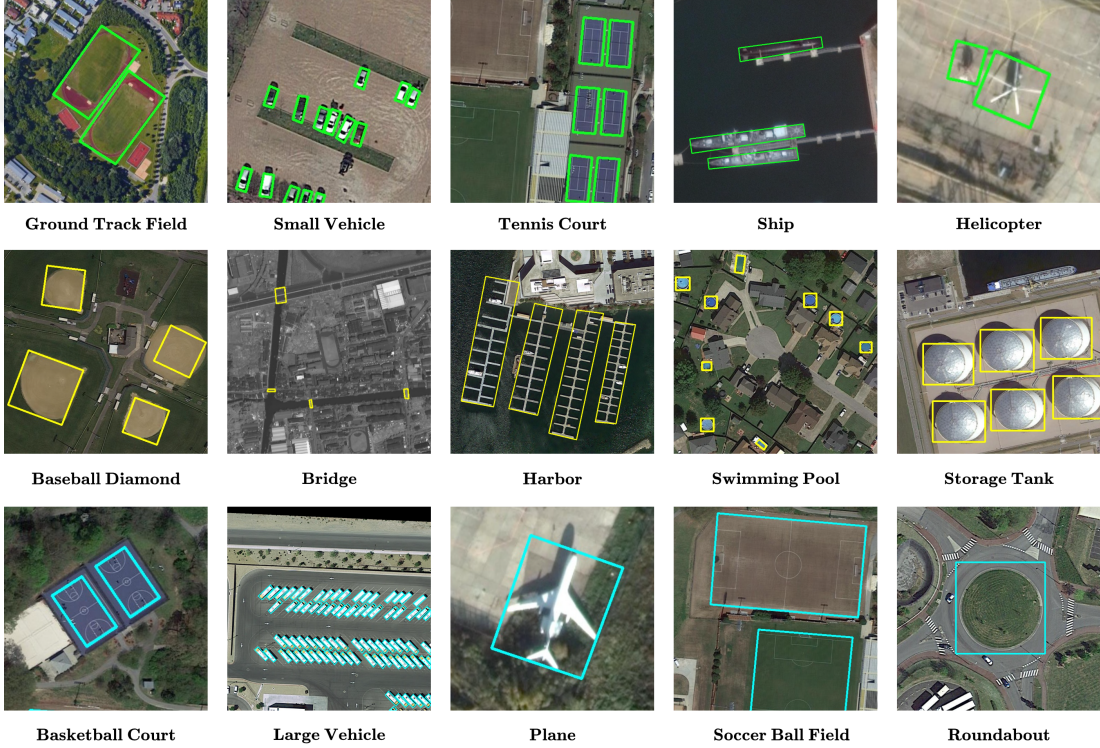


Figure 4.1: Example objects and annotations from DOTA data set with oriented annotations.

Fifteen classes from the DOTA data set with oriented bounding box annotations are illustrated in Figure 4.1. Small and large vehicles in the data set are generally densely placed. Harbors also usually contain ships inside their bounding box areas. The number of training and validation samples are given in Table 4.1. As it can be seen from Table 4.1, both in training and validation sets, the data set is highly imbalanced. The test set information is not publicized, and the evaluation on the test set is done only on the evaluation server that belongs to Wuhan University.

Table 4.1: Class names and number of instances in the data set.

Class name	Training Samples	Validation Samples
Plane	8,055	2,531
Baseball Diamond	415	214
Bridge	2,047	464
Ground Track Field	325	144
Small Vehicle	26,126	5,438
Large Vehicle	16,969	4,387
Ship	28,068	8,960
Tennis Court	2,367	760
Basketball Court	515	132
Storage Tank	5,029	2,888
Soccer Ball Field	326	153
Roundabout	399	179
Harbor	5,983	2,090
Swimming Pool	1,736	440
Helicopter	630	73

4.2 Evaluation Metrics

PASCAL VOC data set [49] is one of the most common natural object detection data sets. Such large-scale data sets launched challenges to advance the research using their data sets and to rank the teams that submit their results, they provided average precision calculation tools to calculate average precision (AP) and mean average precision (mAP) over all object categories.

The average precision is the most commonly used metric to calculate the accuracy of object detection. To be able to calculate the average precision, some additional metrics are used, such as:

- True Positive (TP): Correctly detected object, there is an object in the

image.

- False Positive (FP): Incorrect detection of an object, the detected object does not exist in the image.
- False Negative (FN): No object is detected. However, there is an object or there are objects in the image.

Additional metrics above are also used to populate a table called Confusion Matrix, which is used to visualize the performance of an algorithm. The confusion matrix also includes True Negative (TN), but it does not apply to object detection as there is countless number of objects that are not to be detected within the image.

A threshold is used to determine what is considered true positive or false positive within detection samples. There is an absolute truth information called ground truth which is the correct location of the object. Intersection over Union (IoU) is used to determine the intersection of detection result and the ground truth of the object. If the IoU value is above the selected threshold, it will be considered a correct detection (True Positive), and otherwise, it will be considered an incorrect detection (False Positive).

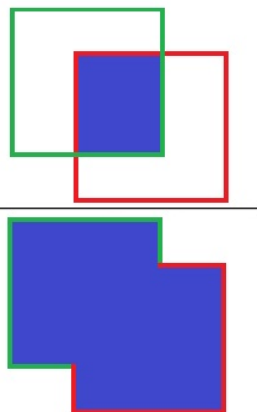
$$IoU = \frac{\text{overlapping area}}{\text{area of union}} = \frac{\text{area of intersection}}{\text{area of union}}$$


Figure 4.2: Intersection over union (IoU).

Some other metrics apart from average precision is Precision and Recall.

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

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{All\ ground\ truths}$$

Average precision is calculated by averaging precision values across all recall values from 0 to 1. The overall accuracy of a detector is calculated by averaging average precision values across all object categories as,

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (4.1)$$

where AP_i is average precision of i^{th} class with N number of classes overall.

Thresholds for IoUs are generally selected and presented with the values of 50%, 75%, and 50% to 95% with increments of 5%. Two other commonly presented metrics are average precision based on the scale of the object (S for Small, M for Medium, and L for Large) and Average Recall (AR).

4.3 Experimental Setup

In this section, we first describe our experimental setup and implementation details for all methods.

Models were implemented in PyTorch, and the model is run on two Nvidia RTX 2080 GPUs with CUDA version of 11.4. Python version 3.8.10 is used with PyTorch version of torch1.8.0+cu111. Each GPU is in charge of the same number of image batches in each run and is set to 4 in our experiments.

We conduct all experiments using two different approaches: (i) horizontal object detection and (ii) oriented object detection. A total of 2,806 images in the DOTA Data set is divided into training set, validation set, and testing set with proportions of 1/2, 1/6, and 1/3 respectively as illustrated in Table 4.2.

Table 4.2: Training/Validation/Test split of DOTA data set.

Training	Validation	Test
1411	458	938

Since the high resolution of images in the data set is in the range of 800x800 to 4000x4000, images are split into 1024x1024 patches with overlaps of 200 pixels. The number of images for training, validation, and test sets are shown in Table 4.3. We use training and validation sets in our experiments to be able to apply parameter changes and see the results faster, which would be a slow process if the evaluation servers for the test set were used.

Table 4.3: Image numbers after image split.

Training	Validation	Test
15749	5297	10833

Table 4.4: Samples after data set split, training and validation set respectively.

Class name	Training Samples	Validation Samples
Plane	16,426	5,298
Baseball Diamond	823	403
Bridge	3,634	897
Ground Track Field	838	316
Small Vehicle	49,864	10,833
Large Vehicle	36,243	9,257
Ship	61,075	19,571
Tennis Court	4,738	1,602
Basketball Court	1,119	297
Storage Tank	9,639	5,027
Soccer Ball Field	849	351
Roundabout	800	299
Harbor	13,085	4,789
Swimming Pool	3,324	763
Helicopter	1,193	134

After the data set split, samples become highly imbalanced in both the training set and validation set as shown in Table 4.4. The number of samples is higher

compared to Table 4.1 because we obtain these images by cropping patches with overlaps of 200 pixels which causes the same object to appear in different contexts or environments in newly obtained patches. For all the experiments, the split given in Table 4.3 is used.

Bochkovskiy *et al.* used mosaic augmentation in their one-stage object detector [42] to increase the detection accuracy of objects outside their context. Regularization techniques to train stronger networks are introduced, such as Mixup [50] or CutMix [51] which only use two images as input however, mosaic data augmentation [42] mixes four training images as shown in Figure 4.3. We employ mosaic data augmentation in horizontal and oriented object detection setups.

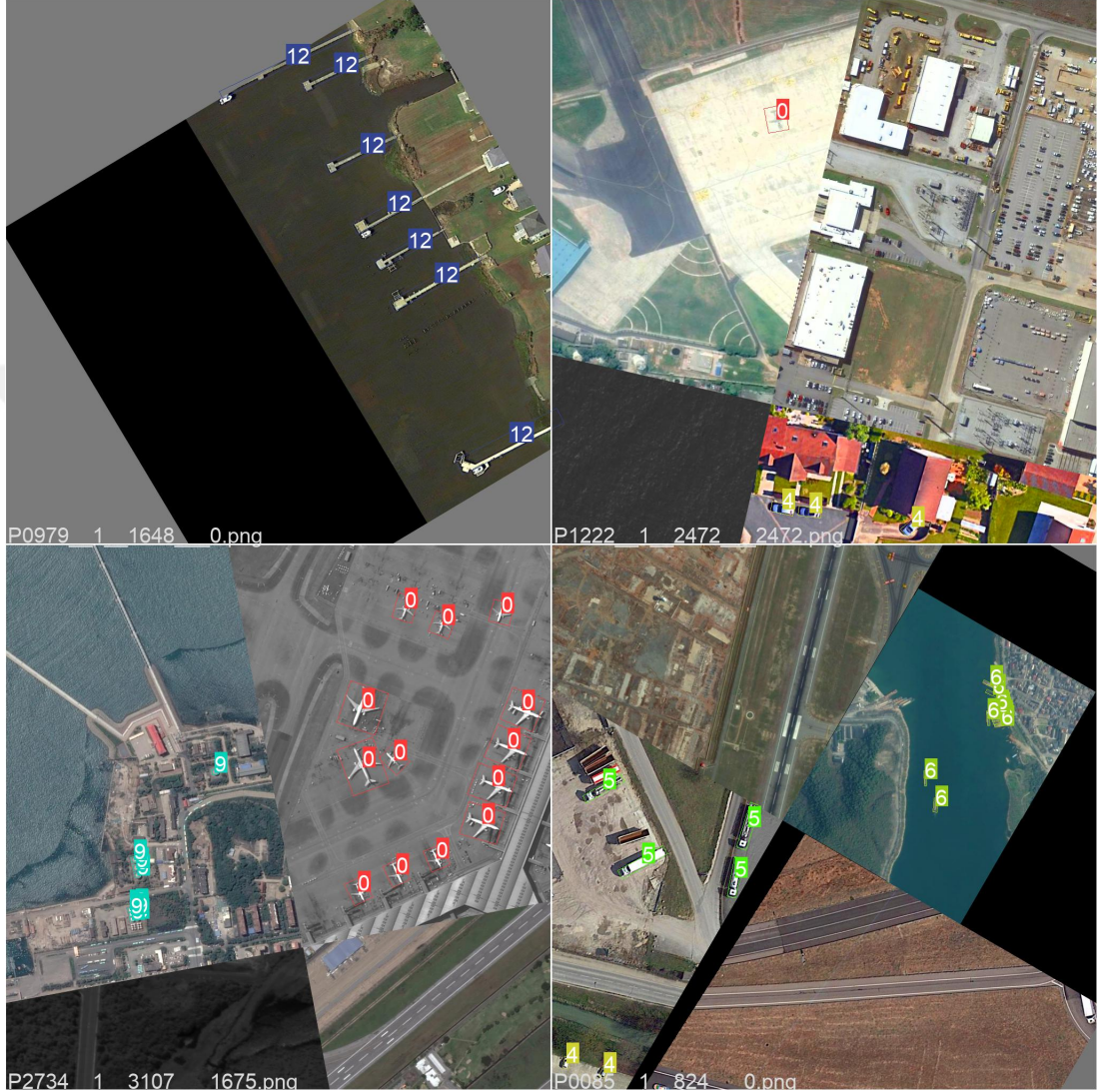


Figure 4.3: Mosaic data augmentation example on a train batch.

4.4 Implementation Details

4.4.1 Improved YOLO Architecture

We train one model for horizontal object detection with our improved YOLO architecture. The model is trained using stochastic gradient descent (SGD) with

10^{-2} learning rate, 0.937 momentum, and 0.0005 optimizer weight decay. During training, the IoU training threshold is set to 0.2, and a batch size of 4 is used. Scaling, adjustments of color space, image flipping, and mosaic data augmentation techniques are some of the affine transformation and data augmentation techniques used on the images. Image HSV-Hue, saturation, and value augmentations are HSV color space augmentations. Image translation, scaling and horizontal image flipping with 0.5 probability are the other augmentation techniques. Image flipping only includes left-right flipping, and mosaic augmentation is applied with 1.0 probability. Model is trained for 140 epochs with these parameters.

4.4.2 Rotated YOLO Architecture

We train one model with different circular smooth label (CSL) parameters for rotated object detection using our improved YOLO architecture with CSL implementation. Similar learning parameters are used for the training phase. The model is trained using stochastic gradient descent (SGD) with 10^{-2} learning rate, 0.937 momentum, and 0.0005 optimizer weight decay. During training, the IoU training threshold is set to 0.2, and a batch size of 4 is used. Scaling, adjustments of color space, image flipping, and mosaic data augmentation techniques are some of the affine transformation and data augmentation techniques used on the images. Image HSV-Hue, saturation, and value augmentations are HSV color space augmentations. Image translation, scaling, and horizontal image flipping with 0.5 probability are the other augmentation techniques. Image flipping for rotated detection includes both left-right and up-down flipping. Mosaic data augmentation with 0.95 probability and Mixup augmentation with 0.1 probability is applied.

We use the Gaussian window function for the circular smooth label and five-parameter bounding box with 180° angular range, which is the long-edge definition. The CSL function includes radius parameter r , which defines the range of the function value depending on the θ parameter of the rotation. We experiment

with radius values of 1, 2, 4, 6, and 8 to see the results of different Gaussian CSL functions. All of the models are trained for 140 epochs with these parameters.

4.5 Results

We stop training at 140 epochs after the validation losses plateau, and there is a decrement observed in the accuracy values.

4.5.1 Improved YOLO Architecture Results

After training for 140 epochs, the validation set mean average precision results are as shown in Table 4.5. The highest accuracy belongs to the classes plane and tennis court. We can observe from the confusion matrix in Figure 4.4 that the classes of soccer ball field and ground track field sometimes get confused by the model. It can also be observed that False Negative (FN) value for helicopter is high, and it is dominantly classified as the plane class. Even though there are high False Positive (FN) values for classes like bridge, soccer ball field, storage tank, swimming pool, and roundabout, these are confused as backgrounds rather than another class object.

Table 4.5: Numerical results (mAP@.5) after 140 epochs of training for horizontal object detection.

Class	mAP @ 0.50
Plane	0.938
Baseball Diamond	0.797
Bridge	0.54
Ground Track Field	0.732
Small Vehicle	0.689
Large Vehicle	0.863
Ship	0.889
Tennis Court	0.947
Basketball Court	0.718
Storage Tank	0.765
Soccer Ball Field	0.548
Roundabout	0.661
Harbor	0.866
Swimming Pool	0.631
Helicopter	0.58
Avg.	0.744

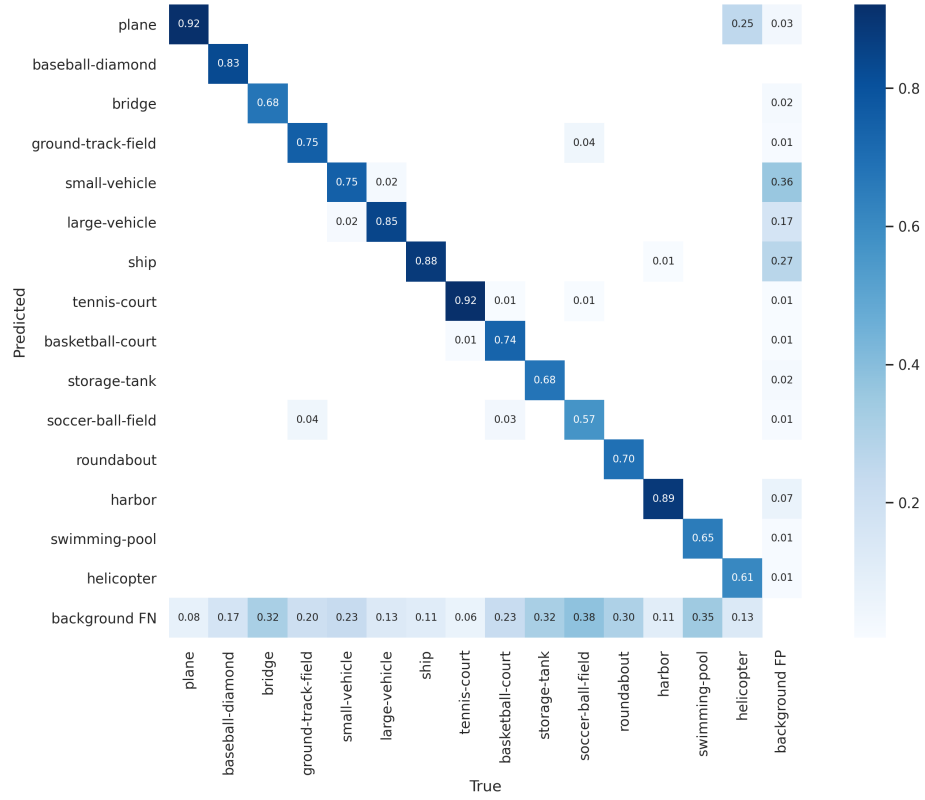


Figure 4.4: Confusion matrix for horizontal object detector.

4.5.2 Rotated YOLO Architecture Results

After training for 140 epochs, the validation set mean average precision results are as shown in Table 4.6 for different radius r values of Gaussian circular smooth label function. It can be observed from the table that plane, tennis court, and ship classes have the highest accuracy among all, and the best result is obtained by the radius value of 1. We can observe from Figure 4.5 that ground track field and soccer ball field objects are classified as each other. The same incident of confusing a helicopter as a plane in horizontal detector also happens in our rotated detector and can be observed from the helicopter column in Figure 4.5.

Table 4.6: Numerical results (mAP@.5) after 140 epochs of training for rotated object detector for different r values in Gaussian CSL function.

Class	$r = 1$	$r = 2$	$r = 3$	$r = 4$	$r = 6$	$r = 8$
Plane	0.98	0.98	0.975	0.975	0.972	0.971
Baseball Diamond	0.814	0.831	0.811	0.811	0.781	0.763
Bridge	0.571	0.549	0.517	0.516	0.498	0.493
Ground Track Field	0.694	0.657	0.667	0.689	0.674	0.659
Small Vehicle	0.81	0.807	0.805	0.812	0.807	0.8
Large Vehicle	0.913	0.91	0.908	0.904	0.903	0.897
Ship	0.967	0.968	0.967	0.968	0.967	0.968
Tennis Court	0.967	0.966	0.962	0.966	0.964	0.961
Basketball Court	0.822	0.838	0.851	0.847	0.845	0.83
Storage Tank	0.811	0.841	0.755	0.652	0.608	0.628
Soccer Ball Field	0.607	0.568	0.538	0.549	0.505	0.566
Roundabout	0.619	0.634	0.367	0.407	0.403	0.341
Harbor	0.857	0.859	0.848	0.85	0.85	0.841
Swimming Pool	0.651	0.58	0.567	0.568	0.589	0.601
Helicopter	0.633	0.56	0.524	0.44	0.494	0.557
Avg.	0.781	0.77	0.738	0.73	0.724	0.725

The confusion matrix for the best rotated detection result on the validation set is given in Figure 4.5.

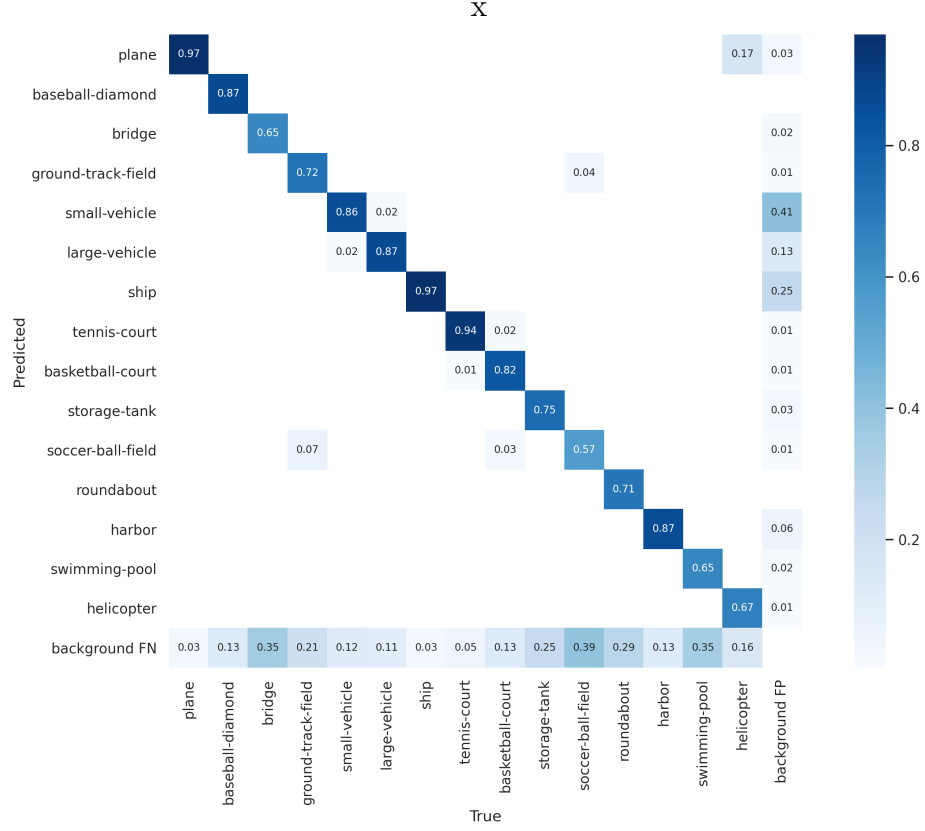


Figure 4.5: Confusion matrix for rotated object detector.

4.5.3 Discussion

It can be seen from the results that by using circular smooth label with our rotated object detector, we can obtain higher accuracy results. Figure 4.6 shows that the CSL function with $r = 1$ radius results in the highest accuracy among our experiments. Followed by CSL with $r = 2$ and *horizontal* detectors we can say that CSL function with higher radius values than $r = 2$ will result in lower mAP.

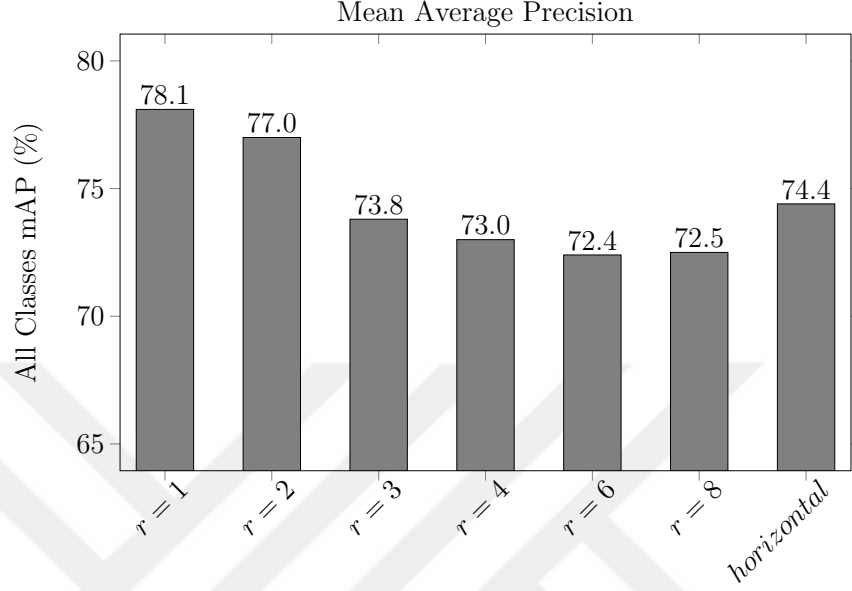


Figure 4.6: Impact of radius (r) on all classes mean average precision (mAP). Radius parameter of circular smooth label (CSL) function results in different labels and thus different accuracies as discussed in Section 3.2.4. The figure also includes the horizontal bounding box detection result.

The comparison of confusion matrices of the horizontal and the rotated detector is a bit uninformative because the majority of mistakes lie in the background class instead of another class in the data set. We only see confusion between the classes of ground track field and soccer ball field and some major confusion between plane and helicopter classes.

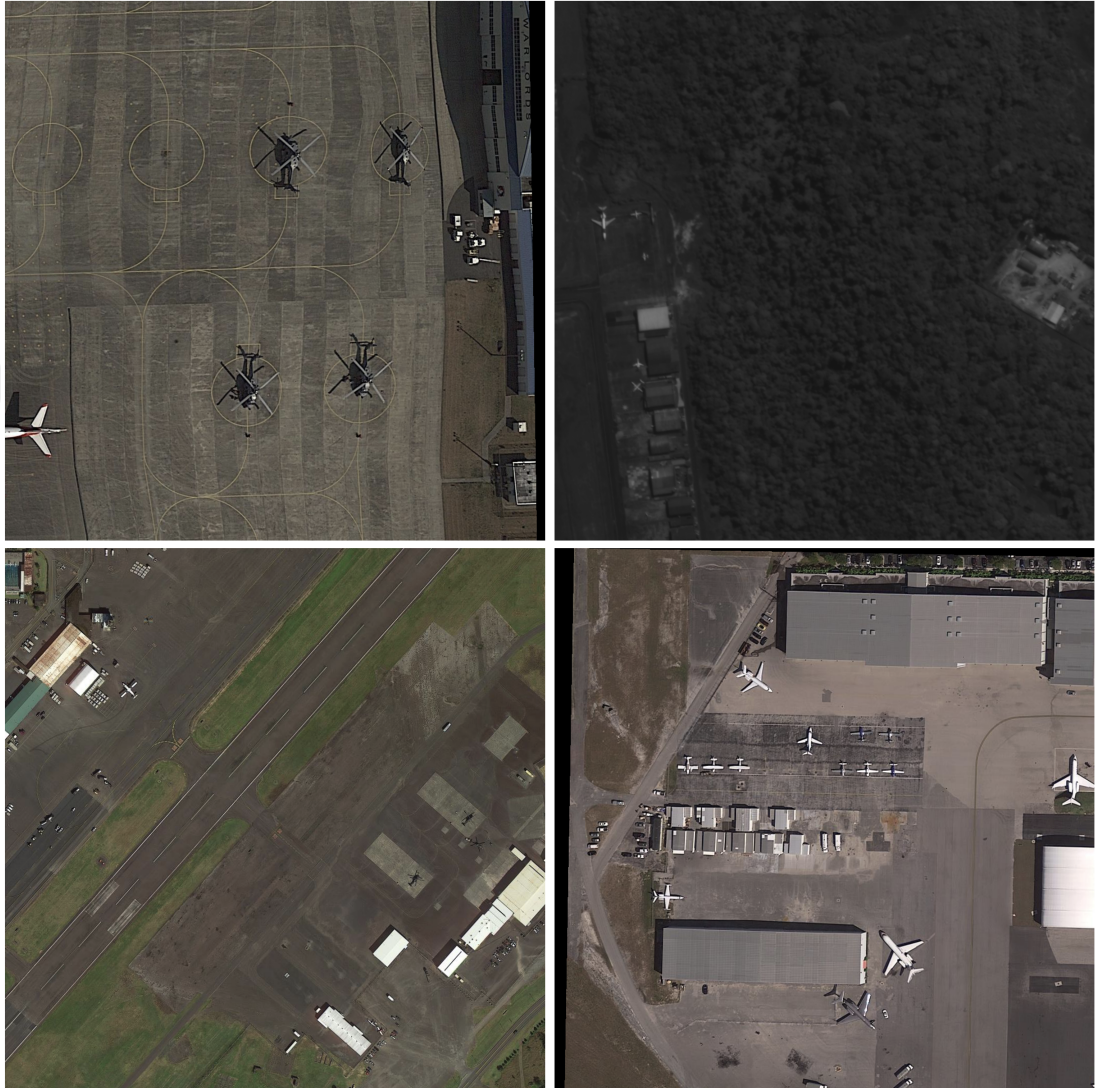


Figure 4.7: Samples from the validation data set that includes plane and helicopter classes together. The left side of the image shows plane and helicopter classes together, while the right side of the image have plane samples that resemble the helicopter class object.

One reason for the high confusion between plane and helicopter classes might be the background information of the two objects, which have similar backgrounds of airfields or airports as shown in Figure 4.7. If we consider air assets to have similar backgrounds, such as airfields, airstrips, airports, and environment around air vehicle hangars, the model is highly biased to detect an air vehicle as a plane

object even though the two classes do not share physical appearance similarity. However, there are similar physical appearances of plane and helicopter classes in some validation samples, and some planes resembling helicopter objects are shown on the right side of the Figure 4.7. There are gray planes and planes that are parked crosswise, which resemble helicopter rotor wings. Another reason for the high False Negative (FN) on the helicopter class is that number of validation samples on helicopter class is 134, and the number of plane samples in validation samples is 5298, which shows an extreme imbalance between the two air asset classes.

Since the boundary problem is more of an issue for the large aspect ratio objects [41], when we compare the accuracy results of these objects for the horizontal and the rotated detector, we expect to see a better performance for the rotated object detector. The classes of bridge, small vehicle, large vehicle, and ship (4-mAP) have higher accuracy results in rotated object detection with circular smooth label which can be observed in Table 4.7.

Table 4.7: Numerical results (4-mAP@.5) for the four large aspect ratio object classes with circular smooth label and horizontal object detectors.

Class	CSL Result	Horizontal Result
Bridge	0.571	0.54
Small Vehicle	0.81	0.689
Large Vehicle	0.913	0.863
Ship	0.967	0.889
Avg.	0.793	0.745

Another observation we make is about the detection results of some particular classes and the number of validation samples belonging to that class in the data set. Plane, small vehicle, large vehicle, ship and harbor objects which have higher number of samples compared to other classes in validation set show robust results for different values of r in the Gaussian circular smooth label setting. An important observation about the relevancy of number of samples and detection results can be made about tennis court class. Even though it does not have the highest

number of samples and has fewer samples than many higher-performing classes, it shares the second-highest performing result with the ship class. The reason for the high performance of the tennis court class is that it has a universal context, and thus the model can learn and distinguish it easily compared to other classes. The tennis court class does not have a very high aspect ratio, and it only includes colors such as orange, blue, and green, and lastly, the structure with lines and environment around the tennis court is the same in every sample of the object. Two samples of tennis court class are shown in Figure 4.8.

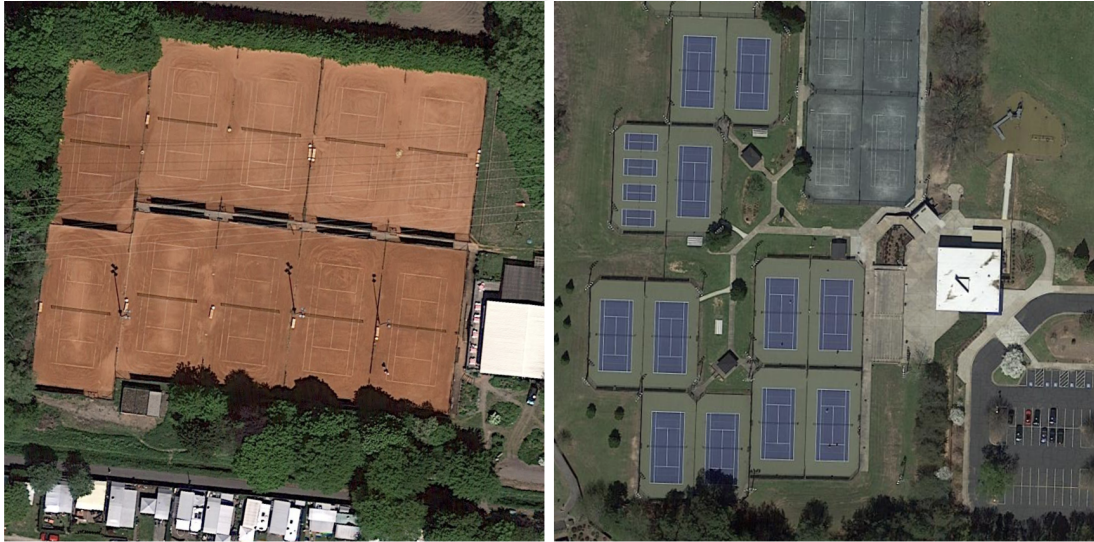


Figure 4.8: Samples from tennis court class.

Lastly, we use DOTA [11] official evaluation development kit to see the results on the validation set. The rotated detection is considered as Task1 of the DOTA remote sensing data set and is evaluated from merged image patches that form the original data set before the image split operation. Results include anchor-free, and two-stage object detection methods and are reported in Table 4.8.

Table 4.8: Evaluation results on DOTA validation set.

Method	mAP ₅₀
CenterNet [52]	63.4
FSM [52]	64.4
DRH-C [52]	64.1
DRH-R [52]	64.1
Faster R-CNN [53]	71.40
Cascade R-CNN [53]	72.76
Cascade R-CNN + LIoU [53]	73.88
Ours	76.95

4.5.4 Qualitative Results

One of the observations from the results is that the only fine-grained class, which is the vehicle with two subclasses that are small and large vehicle is less accurate when subclasses are stacked together. When we check the ground truth labels in Figure 4.9 we can see that both oriented detector and horizontal detector have false positive detections for the small and large vehicle classes.



Figure 4.9: Fine grained class recognition problem in one-stage detection results.

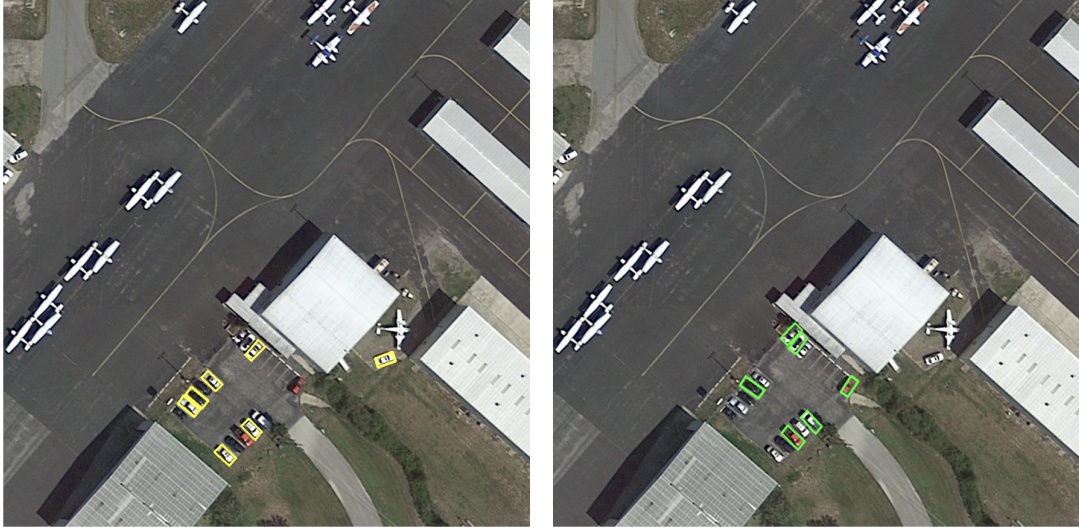


Figure 4.10: Ground truth labels for large and small vehicles from Figure 4.9.

Table 4.7 shows the accuracy difference between horizontal and oriented detection for four large aspect ratio classes. Figure 4.11 shows that in the same settings, the horizontal detector can miss some detections of ship class which are labeled with a dashed red square. Firstly, the circular smooth label is designed to work well in large aspect ratio situations [41]. Secondly, one of the fundamental reasons why oriented object detection is being researched is the condensed placement of objects of interest.



Figure 4.11: Ship detection difference between horizontal and oriented detectors in condensed setting.

Figure 4.12 shows that in the same settings, the horizontal detector can miss some detections of bridge class which is shown in orange colors. The horizontal detector fails to detect the bridge class when there is an extreme aspect ratio for the object of interest. The missed detections are labeled with a dashed red square.

Lastly, the large vehicle and small vehicle classes are the only fine-grained classes that are considered to be among large aspect ratio classes in the data set and in [41]. In some cases, the horizontal object detector fails to detect large vehicles or detect small vehicle class as large vehicle as it can be seen in the upper part of Figure 4.13.

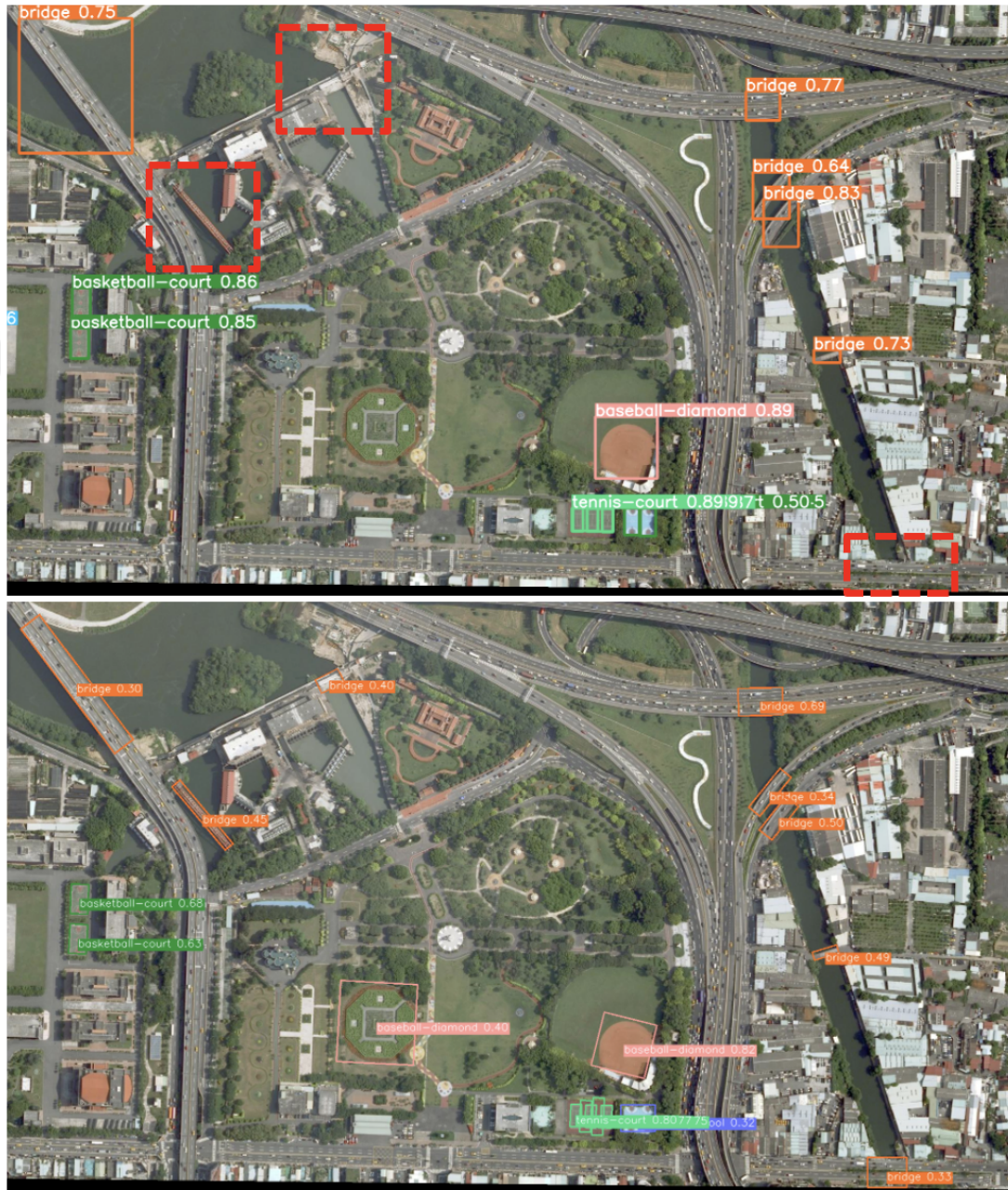


Figure 4.12: Bridge detection difference by horizontal and oriented detectors.

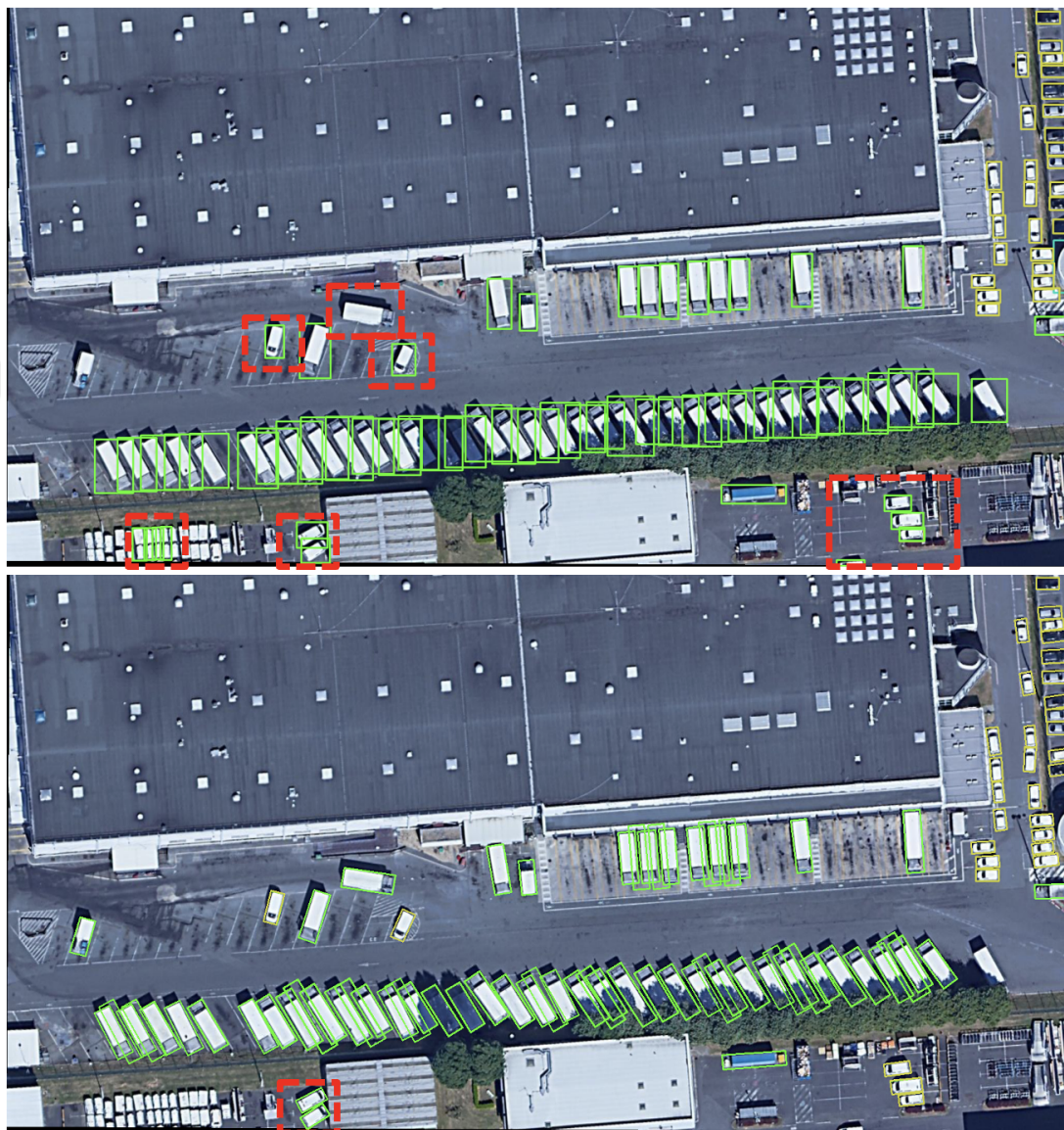


Figure 4.13: Large and small vehicle detection difference by horizontal and oriented detectors.

Chapter 5

Conclusion

We studied a one-stage oriented object detection problem in remote sensing images. While the majority of the applications in the field of remote sensing object detection use two-stage object detectors because of higher accuracy results obtained by them, not many solutions use one-stage object detectors. By using a deeper backbone network, advanced feature fusion techniques, and data augmentation, the one-stage object detector based on the improved YOLO architecture provides comparable results to its two-stage counterparts. We implement an angular classification method called circular smooth label on top of the improved YOLO architecture to be able to detect the objects of interest in an oriented manner. Our experiments confirm that angular classification provides a way to improve accuracy on large aspect ratio objects and also the overall average precision of results on the remote sensing data set. Experiments and results with official data set evaluation provided by the data set authors show that the improved YOLO model with angular classification method achieves higher scores than other methods with a 3.07% improvement over two-stage and 12.84% improvement over anchor-free methods on the task of oriented object detection.

A future work that could follow the experiments and research done in this thesis could be regarding the class imbalance issue after the image partitions in the data set. One could modify the loss function to mitigate such problems

and create a better environment for fewer number samples. Another future work could be to make use of focal loss. Since our work is done via a one-stage object detector and there is a class imbalance issue in general for the one-stage object detectors, we can try to use focal loss instead of binary cross entropy in our experiments. One can also apply the focal loss to the function of circular smooth label. Lastly, instead of angular classification methods, angular regression methods can be implemented to compare the results on the task of oriented object detection.



Bibliography

- [1] R. Schowengerdt, *Remote Sensing: Models and Methods for Image Processing*. Elsevier Science, 2006.
- [2] Y. Yang and S. Newsam, “Bag-of-visual-words and spatial extensions for land-use classification,” in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 270–279, 2010.
- [3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.
- [4] R. Girshick, “Fast R-CNN,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1440–1448, 2015.
- [5] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2881–2890, 2017.
- [6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” in *European Conference on Computer Vision*, pp. 21–37, Springer, 2016.

- [8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
- [9] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *Advances in Neural Information Processing Systems*, vol. 28, pp. 91–99, 2015.
- [10] G. Cheng and J. Han, “A survey on object detection in optical remote sensing images,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 117, pp. 11–28, 2016.
- [11] 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.
- [12] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, “Object detection in optical remote sensing images: A survey and a new benchmark,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 159, pp. 296–307, 2020.
- [13] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European Conference on Computer Vision*, pp. 740–755, Springer, 2014.
- [14] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.” <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>.
- [15] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results.” <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>.
- [16] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Mallocci, A. Kolesnikov, *et al.*, “The open images

- dataset v4: Unified image classification, object detection, and visual relationship detection at scale,” *arXiv preprint arXiv:1811.00982*, 2018.
- [17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014.
 - [18] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, “Selective search for object recognition,” *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.
 - [19] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” *arXiv preprint arXiv:1312.6229*, 2013.
 - [20] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7263–7271, 2017.
 - [21] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International Conference on Machine Learning*, pp. 448–456, PMLR, 2015.
 - [22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
 - [23] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.
 - [24] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125, 2017.

- [25] 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*, pp. 2980–2988, 2017.
- [26] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.
- [27] G. Cheng, J. Han, P. Zhou, and L. Guo, “Multi-class geospatial object detection and geographic image classification based on collection of part detectors,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 98, pp. 119–132, 2014.
- [28] M. Cramer, “The dgpf-test on digital airborne camera evaluation overview and test design,” *PFG Photogrammetrie, Fernerkundung, Geoinformation*, pp. 73–82, 2010.
- [29] H. Zhu, X. Chen, W. Dai, K. Fu, Q. Ye, and J. Jiao, “Orientation robust object detection in aerial images using deep convolutional neural network,” in *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 3735–3739, IEEE, 2015.
- [30] X. Qian, S. Lin, G. Cheng, X. Yao, H. Ren, and W. Wang, “Object detection in remote sensing images based on improved bounding box regression and multi-level features fusion,” *Remote Sensing*, vol. 12, no. 1, p. 143, 2020.
- [31] T. Tang, S. Zhou, Z. Deng, L. Lei, and H. Zou, “Arbitrary-oriented vehicle detection in aerial imagery with single convolutional neural networks,” *Remote Sensing*, vol. 9, no. 11, p. 1170, 2017.
- [32] Z. Liu, J. Hu, L. Weng, and Y. Yang, “Rotated region based cnn for ship detection,” in *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 900–904, IEEE, 2017.
- [33] Z. Zhang, W. Guo, S. Zhu, and W. Yu, “Toward arbitrary-oriented ship detection with rotated region proposal and discrimination networks,” *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 11, pp. 1745–1749, 2018.

- [34] L. Liu, Z. Pan, and B. Lei, “Learning a rotation invariant detector with rotatable bounding box,” *arXiv preprint arXiv:1711.09405*, 2017.
- [35] S. M. Azimi, E. Vig, R. Bahmanyar, M. Körner, and P. Reinartz, “Towards multi-class object detection in unconstrained remote sensing imagery,” in *Asian Conference on Computer Vision*, pp. 150–165, Springer, 2018.
- [36] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.
- [37] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, “Learning roi transformer for oriented object detection in aerial images,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2849–2858, 2019.
- [38] X. Yang, Q. Liu, J. Yan, A. Li, Z. Zhang, and G. Yu, “R3det: Refined single-stage detector with feature refinement for rotating object,” *arXiv preprint arXiv:1908.05612*, vol. 2, no. 4, p. 2, 2019.
- [39] X. Yang, J. Yang, J. Yan, Y. Zhang, T. Zhang, Z. Guo, X. Sun, and K. Fu, “Serdet: Towards more robust detection for small, cluttered and rotated objects,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8232–8241, 2019.
- [40] W. Qian, X. Yang, S. Peng, Y. Guo, and J. Yan, “Learning modulated loss for rotated object detection,” *arXiv preprint arXiv:1911.08299*, 2019.
- [41] X. Yang and J. Yan, “Arbitrary-oriented object detection with circular smooth label,” in *European Conference on Computer Vision*, pp. 677–694, Springer, 2020.
- [42] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” *arXiv preprint arXiv:2004.10934*, 2020.
- [43] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, “Cspnet: A new backbone that can enhance learning capability of cnn,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 390–391, 2020.

- [44] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.
- [45] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8759–8768, 2018.
- [46] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017.
- [47] X. Yang, L. Hou, Y. Zhou, W. Wang, and J. Yan, “Dense label encoding for boundary discontinuity free rotation detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15819–15829, 2021.
- [48] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, “Distance-IoU loss: Faster and better learning for bounding box regression,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 12993–13000, 2020.
- [49] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [50] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “MixUp: Beyond empirical risk minimization,” *arXiv preprint arXiv:1710.09412*, 2017.
- [51] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, “CutMix: Regularization strategy to train strong classifiers with localizable features,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6023–6032, 2019.
- [52] X. Pan, Y. Ren, K. Sheng, W. Dong, H. Yuan, X. Guo, C. Ma, and C. Xu, “Dynamic refinement network for oriented and densely packed object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11207–11216, 2020.

- [53] Y. Zhu, J. Du, and X. Wu, “Adaptive period embedding for representing oriented objects in aerial images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 10, pp. 7247–7257, 2020.

