A COMPREHENSIVE SURVEY ON SMALL OBJECT DETECTION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF
ENGINEERING AND NATURAL SCIENCES
OF ISTANBUL MEDIPOL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
ELECTRICAL, ELECTRONICS ENGINEERING AND CYBER SYSTEMS

By
Melik Ahmet Daye
February, 2021

A Comprehensive Survey on Small Object Detection
By Melik Ahmet Daye
February, 2021

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.

Prof. Dr. Bahadır K. GÜNTÜRK (Advisor)

Asst. Prof. Dr. Muharrem MERCİMEK

Prof. Dr. Hasan F. ATEŞ

Approved by the Graduate School of Engineering and Natural Sciences:

Assoc. Prof. Dr. Yasemin Yüksel Durmaz
Director of the Graduate School of Engineering and Natural Sciences

Foreword

This master thesis is the mark of the accomplishment for master study in Electrical-Electronics Engineering and Cyber Systems program at Istanbul Medipol University.

During my master study, various topics had been chosen to research and develop novel ideas to become a master thesis. Each chosen topic create opportunity to gain knowledge and experience. Then, I realized that accessing all bindings of topic requires devoted hard work. From that point, I decided to prepare comprehensive and comparative study on small object detection. The aim of this thesis is introducing collection of knowledge about small object detection may guide and facilitate the further studies.

To bring up important concepts of this topic and make smooth connection between them has not been easy during research but complete survey has been merged at the end. I would like to express my gratitude to my supervisor Prof. Dr. Bahadır K. Güntürk. His valuable ideas and guidance make me accomplish the research and finalize this thesis.

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: MELIK AHMET DAYE

Signature :

Acknowledgement

I would like to express my sincere appreciation my thesis advisor, Prof. Bahadir K.Gunturk who always helpful and instructive about my research or writing. He shows support in everything I do and enlighten me the with his knowledge for my thesis writing.

Finally, I express my gratitude to my family and friends for providing me with full support and solid encouragement entire my master duration and phase of researching and writing for this thesis. They have significant contribution for this achievement.

Contents

List of Figures

List of Tables

List of Symbols

$\boldsymbol{AP_S}$ Average precision of small objects

$\boldsymbol{AP@0.5:0.95(S)}$ Average precision of small objects between interpolated intersection of union ratio of 0.5 and 0.95

$\boldsymbol{AP@0.5:0.95}$ Average precision of objects between interpolated intersection of union ratio of 0.5 and 0.95

$\boldsymbol{AP@0.5}$ Average precision of objects equal and above the intersection of union ratio of 0.5

$\boldsymbol{AP@0.75}$ Average precision of objects equal and above the intersection of union ratio of 0.75

Abbreviations

BFEN Backward Feature Enhancement Network. 49

CFE Comprehensive Feature Enhancement. 43, 69, 70

CNN Convolutional Neural Network. 1, 7–9, 11, 36–38, 48

CPU Central Processing Unit. 73

DCNN Deep Convolutional Neural Network. 13

DOTA Dataset for Object Detection Aerial Images. 27

DPM Deformable Part-based Model. 4, 5, 9, 35

DSSD Deconvolutional Single Shot Detector. 28, 42, 43, 63, 70

ERF Expanding Receptive Field. 45

FAIR Facebook AI Research. 8

FFM Feature Fusion Module. 44

FGRM Feature Guided Refinement Module. 42, 44

FPN Feature Pyramid Network. 9, 12, 34, 39, 41, 44, 46–48, 57, 71, 72

FPS Frames Per Second. 14, 64, 71, 72, 77

FSAF Feature Selective Anchor Free. 56, 57

FSSD Feature Fusion Single Shot Detector. 28, 43, 63, 70

GAN Generative Adversarial Network. 53, 54

GPU Graphics Processing Units. xix, 1, 8, 35, 64, 73

HOG Histogram of Oriented Gradients. 4, 35

IOD Insertion of Detection. 26

ION Inside-Outside Net. 52, 72

IOU Intersection Over Union. 22–25, 56

LSTM Long Short-Term Memory. 52

mAP Mean Average Precision. 24, 25, 29

MOT Multiple Object Tracking. 27

MS-COCO Microsoft Common Objects in Context. 8, 15, 19, 22, 24–26, 29, 30, 34, 56, 64, 72, 73

MSCF Multi Scale Contextual Features. 41, 44

PAN Path Aggregation Network. 46

R-CNN Regions with CNN. 9–12, 14, 30, 34, 46, 48, 51, 52, 54, 69, 71, 72, 77

ROI Region of Interest. 11

RPN Region Proposal Network. 11, 13, 49, 59

SFAM Scale-wise Feature Aggregated Module. 44

SIFT Scale-invariant Feature Transform. 4

SLPN Spatial Layout Preserving Network. 49

SNIP Scale Normalization for Image Pyramids. 38

SNIPER Scale Normalization for Image Pyramids Efficient Resampling. 38, 72

SOD-MTGAN Small Object Detection via Multi-Task Generative Adversarial Network. 54

SOS Small Object Sensitive. 38

SOT Single Object Tracking. 27

SPM Spatial Pyramid Matching. 11

SPP Spatial Pyramid Pooling. 11, 46, 69

SSD Single Shot Detector. 14, 15, 28, 29, 38–40, 42–44, 54, 63, 69, 70, 77

SURF Speeded Up Robust Features. 4

SVM Support Vector Machine. 3, 11

TUM Thinned U-shape Feature Aggregation Module. 44

UAV Unmanned Aerial Vehicle. 45

VGG Visual Geometry Group. 7, 8, 42, 48, 69–71

VJ Viola-Jones. 4

VOC Visual Object Classes. 5, 22, 24, 25, 29

VSSA Vertical Spatial Sequence Attention. 52

ÖZET

KÜÇÜK NESNELERIN TESPİTİ ÜZERİNE DETAYLI İNCELEME

Melik Ahmet Daye
Elektrik-Elektronik Mühendisliği ve Siber Sistemler, Yüksek Lisans
Tez Danışmanı: Prof. Dr. Bahadır K. GÜNTÜRK
Şubat, 2021

Yapay zeka ve bilgisayarlı görü sistemlerindeki gelişmeler nesne tanıma teknolojisini bilgisyarlı sistemler için kolaylaştırmaktadır. Nesne tanıma teknolojisi yüz tanıma, güvenlik sistemlerinde, robotik sistemlerde, sürücüsüz arabalarda ve benzeri sistemlerde kapsamlı bir şekilde kullanılmaktadır. Nesne tanıma teknolojisi üzerinde yapılan çalışmalar görüntülerden nesneyi tanımlayan özelliklerin toplamasını geliştirerek, nesnelerin sınıflandırmasını ve görüntedeki yerlerinin belirlenmesinin doğruluğunu artırmayı hedeflemektedir. Geleneksel metotlar manuel olarak belirlenen tanımlayıcı özellikler ve filtrelemeler ile nesneleri tespit algoritmaları ile bulmaktadır. Grafik işleme üniteleri ile evrişimli sinirsel ağların gelişiminde sonra, nesneyi tanımlayan özelliklerin çıkartılması hangi özelliklerin daha önemli olduğunun algoritma tarafından otomatik bir biçimde öğrenilmesiyle nesne tespiti sistemlerinin başarım oranı artmıştır. Büyük veri kümeleri ve derin öğrenme ağları ile nesne tanımanın başarı oranı artmakta ama buna rağmen bu sistemlerin nesne tanıma üzerinde eksikleri bulunmaktadır. Bu eksiklerde bir tanesi küçük sayılan nesnelerin bulunmasındaki zorluktur. Bu zorluğa sebep olan nedenlerin tespiti ve bunlara karşılık gelebilecek çözümler üzerine bir çok araştırma yapılmaktadır. Bu çalışmamızda küçük sayılan objelerin tespitini zorlaştıran etmenleri ve bunlara uygun çözümleri bugüne kadar yapılan önemli çalışmalar üzerinden detatylı bir şekilde sunacağız. Bu çalışmalardaki kullanılan çözümlerin metodolojisi ve birbirlerine göre mukayesesini de boşarım oranı ve verimliliği üzerinden yorumlayacağız.

Anahtar sözcükler: Derin öğrenme, nesne tespiti, küçük objeler, inceleme, karşılaştırma, algı alanı, nesne özelliklerinin gösteriminin geliştirilmesi, veri artırımı, görüntü parçalarını kullanma, çoklu ölçekte nesne özellikleri, hiper parametre optimizasyonu, bağlamsal özellik kullanımı, süper çözünürlük.

ABSTRACT

A COMPREHENSIVE SURVEY ON SMALL OBJECT DETECTION

Melik Ahmet Daye
M.S. in Electrical, Electronics Engineering and Cyber Systems
Advisor: Prof. Dr. Bahadır K. GÜNTÜRK
February, 2021

The advancement in artificial intelligence and computer vision facilitates object detection for computer-based systems. Object detection is used in a wide spectrum of applications such as face detection, enforcement application, robotic vision, autonomous cars, etc. The studies on object detection aim to improve feature extraction from images and improve the classification and localization of objects in an image. The traditional methods achieve object detection via handcrafted features or filters with prediction algorithms. After the improvement of convolutional neural networks and stronger processing units such as Graphics Processing Units (GPU), the features extracted in a way of that which features are important to predict object information more precisely. The huge datasets and deeper networks are used to increase accuracy in object detection but every system has some drawbacks. One of the drawbacks of object detection has the difficulty to detect small objects in images. Many types of research were conducted on the reasons for the challenge in small object detection and important approaches are developed the solve this problem. In our thesis, we explain the drawbacks and solutions in the light of prominent studies comprehensively. Also, we discuss the methodology of solutions and compare them in terms of accuracy and efficiency.

Keywords: Deep learning, object detection, small objects, survey, comparison, receptive field, feature enhancement, augmentation, image tiling, multi-scale features, optimizing hyper-parameters, contextual reasoning, super-resolution.

Chapter 1

Introduction

The computer vision field has been developed through the years for computers to perceive the world similar to human vision systems. The human vision system consists of eyes and a complex neural brain system to interpret the signals from the eyes. In parallel to the human vision system, the researchers have been developing several types of sensors similar to the eyes and algorithms similar to the human brain. Object detection is crucial technology in the computer vision field because recognizing and locating objects is an essential ability to interpret the world for various applications such as reading license plates, interpret the situations from images, face detection, and recognition, autonomous cars, facilitating enforcement activities, etc. Early object detection systems are made by hand-crafted features from images and prediction algorithms. After the advancement in GPU, Convolutional Neural Network (CNN) based algorithms have been developing to converge the human neural systems. In recent years, CNN based object detectors achieve remarkable success to classify and localize objects which have different visual appearances in various environment

Despite progress in object detection, object detectors have some weaknesses against some challenging cases such as small objects. Small objects are hard to detect in images because of their inadequate visual appearances. They can be affected by occlusion, low-resolution imaging, and imbalance between other classes.

In our work, we examine the reasons for challenges in small object detection and gather the proposed techniques that present novel ideas to overcome challenges. Each technique is explained in detail and supported with relevant studies. A comparison is also provided to show the effectiveness of approaches.

In Chapter 2, the general object detection techniques are explained in detail. In Chapter 3, we describe the small objects and explain reasons which cause a challenge in small object detection. In Chapter 4, we explain the approaches that improve small object detection with their example proposed methods and make connections between reasons of challenge and proposed solutions. The evaluation results of methods that are explained in Chapter 4 are shown in Chapter 5 and we compare the methods in detail in terms of accuracy and efficiency. In Chapter 6, we conclude our work and mention future work.

Chapter 2

Object Detection

Object detection is a technology field of computer vision and image processing that wants to achieve human-like recognizing and identification of objects in images. The main goal of this technology is localizing objects and making decisions about their classes using visual features from digital images. Object detection has a huge impact on developing text recognition, image annotation, activity recognition, face detection and recognition, robotic vision, autonomous cars, surveillance and reconnaissance applications, etc. The general approaches for this task are divided into two main categories; traditional approaches and deep learning approaches.

2.1 Traditional object detection

Traditional approaches are based on manually designed and extracted features and more simple machine learning techniques which are Support Vector Machine (SVM), decision trees, random forest, etc. The common architecture of traditional techniques can be separated into three stages: proposal generation, feature extraction, and classification [9]. The proposal generation is based on searching regions of interest that contain objects, this search can be accomplished with the

sliding windows method [12, 13]. The feature extraction stage is responsible for obtaining feature vectors and encoded by feature descriptors such as Histogram of Oriented Gradients (HOG) [14], Haar [15], Scale-invariant Feature Transform (SIFT) [16] or Speeded Up Robust Features (SURF) [17]. The last stage, classification, learns to match labels of object classes to the regions which are proposed and encoded with feature descriptors.

2.1.1 Viola-Jones detectors

P. Viola and M. Jones developed the first real-time object detection method which specialized in face detection [18, 19]. Viola-Jones (VJ) detector is a combination of different methods which are Haar [15] features, integral image, Adaboost [20] algorithm and detection cascades. VJ detector uses a sliding window approach to find faces in an image with the help of Haar features and Adaboost algorithm which is used for feature selection. The real-time performance comes with an integral image, feature selection, and detection cascades methods [10].

2.1.2 HOG detector

N.Dalal and B.Triggs made significant improvements on SIFT [16] and proposed a new feature descriptor named HOG [14]. It uses different scales of images for detection without changing detection windows size and overlapping local contrast normalization blocks to make a robust detector for feature in-variance(translation, scaling, illumination, etc.) and the non-linearity [10]. It is used primarily for pedestrian detection.

2.1.3 Deformable part-based model

Deformable Part-based Model (DPM) was developed on top of the HOG detector by P.Felzenszwalb in 2008 [21], and R.Girschick made important contributions

to the DPM detector. The DPM detector detects objects with Markov random fields which use the "divide and conquer" technique to produce mixture graphical models of objects from different parts of objects as deformable parts. The DPM contains three major parts: root filter, part filters, and spatial model. Root filter extracts more shallow features from the whole object, part filters learns features from smaller parts of the object more extensively than root filter and spatial model makes decisions about locations of the part filter for the root. The filters can learn their configurations with a manner of weakly supervised learning as latent variables. R.Girshick developed some important techniques for improving the accuracy of DPM detectors such as hard negative mining, bounding box regression, and context priming. Later on, these methods are used by modern object detection techniques with new improvements for many years. The DPM detector was the winner of Visual Object Classes (VOC)-07, -08, and -09 detection challenges, P. Felzenszwalb and R. Girshick also were awarded the "lifetime achievement" by PASCAL VOC [10].

2.2 Object detection with based on deep learning

Deep learning approaches are based on feature extraction backbones which mostly consist of convolutional layers that have the capability of learning high-level representation of important image features and modern object detector parts. Most deep learning-based object detection techniques are composed of three common stages: an input layer, backbone, and prediction part. Also, some network architectures contain neck parts that collect feature maps from different stages of the network. Furthermore, object detection techniques are divided into two groups with respect to their prediction parts: one-stage detectors and two-stage detector. The common pipeline of both one-stage and two-stage detectors is shown in Figure 2.2.1. The pipeline begins with feeding input into a backbone network, then the output of the backbone network is named as feature maps. The neck part follows the output backbone but it is optional and is used to increase the quality of features in features maps. The last stage is the prediction stage and it differs between one-stage and two-stage detectors. In the next section, we will

describe the backbone and introduce some of the preeminent backbone networks which are used commonly in deep learning. Then, we will describe the meaning of one-stage and two-stage detector and examine them. Finally, we will review some important deep learning-based object detection techniques chronologically. The neck parts of object detectors are not explained in this section but we will give some examples and we will mention them in the next sections when we will address the solutions for small object detection.

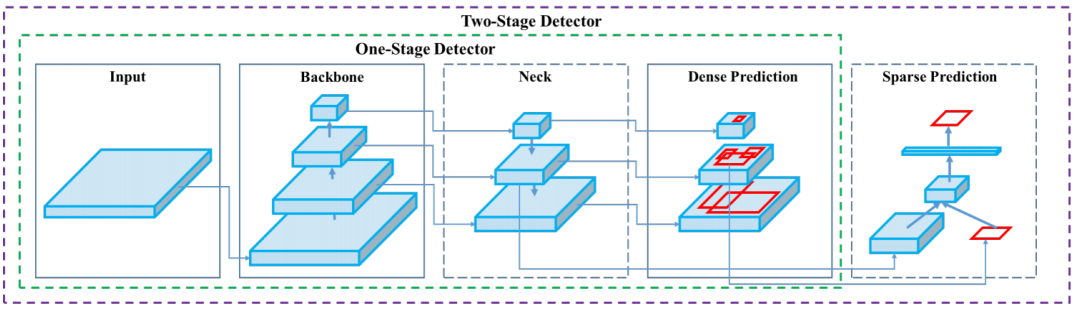


Figure 2.2.1: The common pipeline of deep learning based object detector [1].

2.3 Backbone networks

Backbone networks act as hand-crafted features of traditional methods but the main difference is they can learn and extract features from training data and generate a high-level representation of images, which are called feature maps. The carefully selected backbone network can directly affect the memory, speed, and performance of the detector [22]. The convolutional layers perform as feature extractors by applying a filter to the input image and they are critical layers of the backbone network. The backbone networks usually are trained on ImageNet [23] or OpenImage [24] dataset.The design of the backbone network should meet the trade-off between accuracy and speed. The real-time applications may require more efficient and shallow backbone networks. On the other hand, if accuracy is more important deeper and more complex backbone networks can be preferred. The most famous backbone networks can be named as : AlexNet [25], VGG [26], GoogLeNet [27], ResNet [28], Darknet19 [29], Darknet53 [30] , ResNeXt [31],

Xception [32] and etc.

2.3.1 AlexNet

AlexNet [25] was the winner of the ImageNet ILSVRC-2012 competition and outperformed the traditional methods. The network that has eight layers was considered the first large scale CNN.

2.3.2 VGG

The Visual Geometry Group (VGG) [26], a famous backbone network, was introduced by Oxford VGG in 2014 and has two most common types VGG-16 and VGG-19. The VGG network is deeper than AlexNet [25] and uses smaller convolution filters than AlexNet. It still has been used widely in detection networks since proposed.

2.3.3 GoogLeNet(Inception)

Google Inc. proposed this deeper architecture at first in 2014, then some enhancement has come through the years and Inception has 4 versions. The factoring convolution and batch normalization techniques were introduced by Inception [27] network.

2.3.4 ResNet

The Deep Residual Networks [28] was introduced as a very deep network by K.He et al. in 2015 and it may have up to 152 layers. The most known types are ResNet50,ResNet101 and ResNet152. The contribution of ResNet

to CNN is residual layers which can feed by previous layers feature and making easier the training of networks. ResNet was the winner of ImageNet [23] detection-localization and Microsoft Common Objects in Context (MS-COCO) [33] detection-segmentation when introduced in 2015.

2.3.5 Darknet-19

Darknet-19 was proposed with YOLOv2 [29] by Joseph Redmon and Ali Farhadi in 2016. The design of the network was inspired by GoogLeNet [27]. The design of convolutional layers filters are similar to VGG [26] networks, smaller filters such as 3x3 and 1x1 were used. Also, batch normalization is used to stabilize and accelerate training. Darknet-19 has 19 convolutional layers and 5 max-pooling layers. It achieved 72.9% top-1 accuracy and 91.2% top-5 accuracy on ImageNet [23]. It has real-time performance with 5.58 billion operations for inferencing to an image.

2.3.6 Darknet-53

Darknet-53 was introduced as a backbone for the YOLOv3 [30] object detection network in 2018. The important differences between Darknet-53 and its predecessor Darknet-19 [29] are usage of residual connections like ResNet [28] networks.Also, Darknet-53 is deeper than DarkNet-19 with 53 layers and it still has real-time inference with successive performance in utilizing GPU.

2.3.7 ResNeXt

The ResNeXt [31] was introduced by Facebook AI Research (FAIR) as the next generation of ResNet [28] architecture in 2017. The ResNeXt has aggregated transformations, group convolutions, and cardinality dimensions which makes the network wider and more complex than ResNet.

2.3.8 Xception

The lighter version of Inception [27] network with the usage of depthwise separable convolutional layers. It was introduced in 2017 by Google Inc. It is the backbone of many object detectors such as MobileNet [34] and LightHead Regions with CNN (R-CNN) [35]. It is very suitable for deep learning tasks that operate in embedded systems because of requiring low power and computation cost.

2.4 Two-stage detectors

The two-stage detectors have to produce a set of proposals which are potential regions of interest that may contain objects. In the first stage, region proposals are generated to further refining of class and localization information in the second stage. In the second stage, features and pre-prediction information are used to obtain final prediction results. The first examples of deep learning-based object detection techniques are designed as two-stage detectors. The performance of traditional object detection techniques and handcrafted features has reached a plateau after 2010 [10]. When CNNs became popular after 2012 , R. Girschick et al., the developer of DPM [21], proposed a new detector: R-CNN [36] for object detection [10]. The R-CNN is the first object detector based on deep learning and a two-stage detector. The first two-stage detectors use sliding windows techniques such as Selective Search [37]. The first detectors have trainable and untrainable parts. Thus, the two-stage detectors do not work in an end-to-end fashion and they are slow because of searching algorithms. Then neural network-based region proposal networks are used to generate proposals and they become an end to end trainable and faster. Moreover, most two-stage detectors make predictions on a single feature map that degrade the performance of the detector because detectors are not sensitive to objects in different scales. The Feature Pyramid Network (FPN) was proposed to solve single-scale issues with multi-scale feature pyramid and path aggregation.Figure 2.4.1 illustrates the common structure of two-stage detectors.

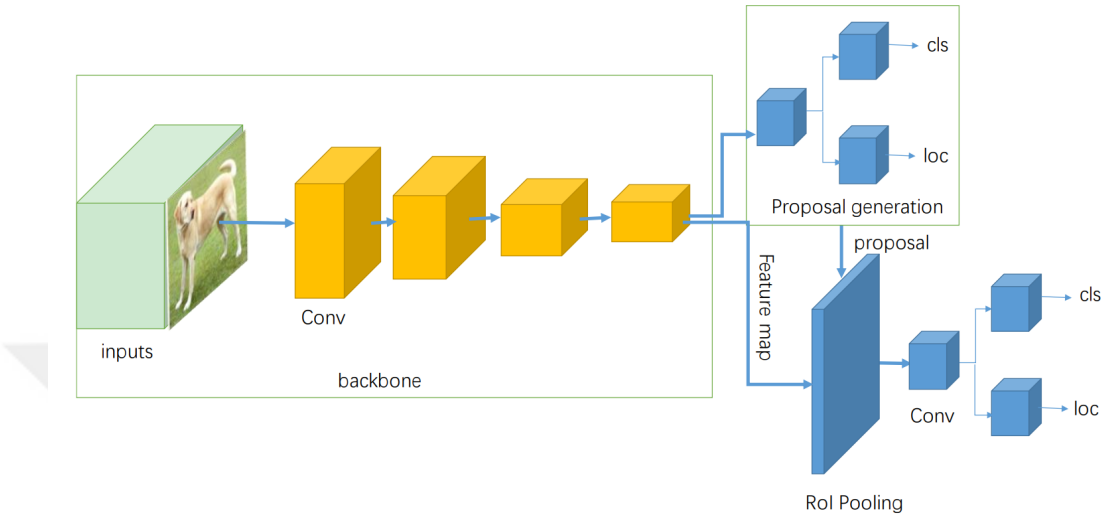


Figure 2.4.1: The common structure of two-stage object detectors [2].

2.4.1 R-CNN

R-CNN [36] is one of the two-stage object detector methods that has improved the performance of object detection. This proposed method has three significant parts which are proposal generation, feature extraction, and region classification. By using Selective Search [37], R-CNN is able to create a sparse set of propositions which is intended to dismiss regions that can undoubtedly be recognized as background regions for every image. Each proposition is cropped and resized for the same size in order to be encoded into a feature vector. By these extracted features, bounding box regression is able to learn the unique proposals of the object. R-CNN uses a transfer learning method which uses pre-trained weights of ImageNet [23] except for the last fully connected layer so the R-CNN architecture is fine-tuned on the pre-trained model and by this way the performance of the model is increased significantly. Using this proposed method produces more robust and discriminative features for classification when it is compared to traditional methods.

2.4.2 SPP-Net

Spatial Pyramid Pooling (SPP)-Net [38] aims to extract more descriptive features using the Spatial Pyramid Matching (SPM). The special layer called spatial pooling layer computes the feature map from the entire picture utilizing the extractable same size feature vectors. In this way, there are no cropping proposal regions to feed into the CNN model like in R-CNN so there is no information loss and it can obtain more information from different scales. Thus, extracted features are given to bounding box regression and SVM classifier. SPP-Net is not only getting more descriptive features, it is also faster compared to R-CNN [36].

2.4.3 Fast R-CNN

Fast R-CNN [39] focuses on the two negative sides of SPP-Net [38] which are needs for extra cache memory to store extracted features and limiting the learning capability of backbone architecture of SPP-Net. However, Fast R-CNN computes the feature map like SPP-Net but uses an Region of Interest (ROI) pooling layer to extract regional features by taking a single scale partition of the proposals into the same number of divisions and this crucially affects the backbone of SPP-Net's learning capability and memory caching. Hence, all the steps are optimized end-to-end and achieve better performance and inference speed compared to R-CNN and SPP-Net.

2.4.4 Faster R-CNN

Faster R-CNN's [40] one of the unique ideas was using a proposal generator which is named as Region Proposal Network (RPN). This newly developed network can learn a high level of visual information and be able to create new information about the region in a data-driven manner. Even though Faster R-CNN corrects the speed bottleneck of Fast R-CNN [39], the issue of detecting small objects at different scales was still difficult with Faster R-CNN because of the single deep

layer feature map in order to predict the last results.

2.4.5 FPN

FPN [41] tries to increase the low performance of Faster R-CNN when detecting objects at different scales. The proposed network is designed on top of Faster R-CNN [40] with FPN block. FPN block uses hierarchical feature maps instead of using a single feature map and it merges the deep layer features with a less deep layer to be able to detect objects at different scales. The semantic-rich features are shared to shallow feature maps from deeper maps via lateral connections. The feature pyramid networks achieve great performance with respect to Faster R-CNN. The FPN module has been used by many recent detectors and the classical structure of FPN has developed and modified for better performance through the years.

2.5 One-stage detectors

The one-stage detectors are designed to eliminate the region proposal phase of two-stage detectors and offer much faster detectors to meet up requirements of a real-time application without sacrificing accuracy too much. The one-stage detectors treat an image as a whole potential area contains an object and try to predict coordinates of bounding box and class of object concurrently. The precision of early one-stage detectors can be lower than two-stage detectors but they have an advantage of higher inference speed. The first single-stage detectors, OverFeat [42] and YOLO [43] predict bounding box and class of objects with regression on a single feature map and they do not use any priors such as anchor boxes. Then, multi-scale feature maps and anchor boxes are used to boost the performance of one-stage detectors and their performance becomes comparable with two-stage detectors. With further developments, one-stage detectors have become more popular, robust, and outperform two-stage detectors in terms of accuracy and inference speed. Figure 2.5.1 illustrates the common structure of

one-stage detectors.

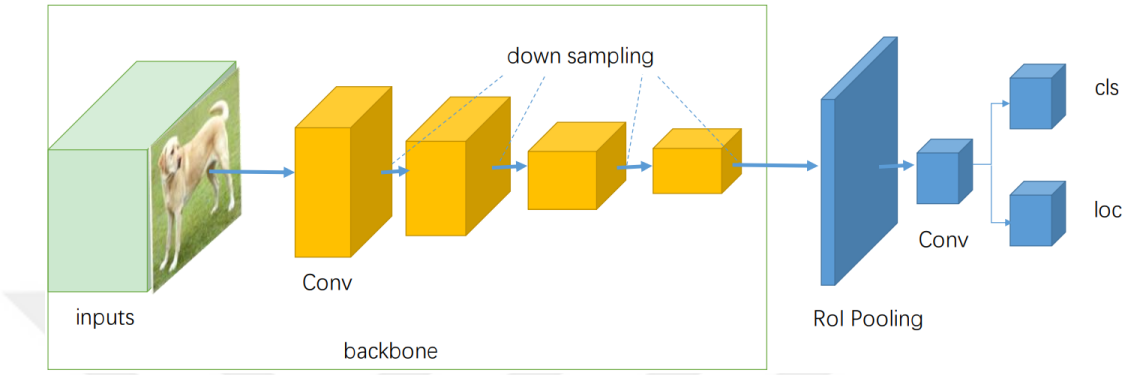


Figure 2.5.1: The common structure of one-stage object detectors [2].

2.5.1 OverFeat

OverFeat [42] is considered the first successful one-stage detector that was proposed by Sermanet et al. The OverFeat handles object detection as a multi-region classification problem. The classifier part uses features from the output of Deep Convolutional Neural Network (DCNN) to detect potential objects and their classes in regions. The bounding-box regressor part uses these predicted regions and the same features from DCNN to calculate the localization of objects precisely. The image pyramid paradigm is used to achieve robust multi-scale detection. Images at different scales are fed into the network and results are merged to show refined detections.

2.5.2 YOLO

The YOLO series has been developed over the years as a one-stage detector. The root detector of the series, YOLO, was developed by Redmon et al [43] as a real-time detector. YOLO is not designed to produce proposals like RPNs and use fixed proposals which are each region of an image when divided into a 7x7 grid. The network is configured to detect two objects for each grid using regression.

The regression predicts the location and class of objects. The inference speed of YOLO is very high when compared to other object detectors in the same era and reach 45-155 Frames Per Second (FPS) with different backbones. However, YOLO has some drawbacks despite high-speed inference. First, there is a limitation on the number of objects that can be detected for each grid so the network cannot achieve dense predictions. The second disadvantage is the lack of support for multi-scale features because of the single feature map. Thus, it produces poor detection performance for objects at different scales and sizes.

2.5.3 SSD

After the publication of YOLO [43], Liu et al developed a one-stage detector named Single Shot Detector (SSD) [5] which tries to fix the inadequacies of YOLO. The first remarkable change of SSD over YOLO is predictions are done on multi-scale feature maps. The feature maps at different scales are collected from different layers of the network to detect objects in different scales and aspect ratios. The shallow layers have rich spatial information to facilitate the detection of small objects. The grid system of SSD is similar to YOLO but each grid is associated with predefined anchor boxes that generalize the scale and aspect ratio of the given dataset. The location of objects is predicted by regression with obtaining offsets for each bounding box from predefined anchor boxes. The weighted summation loss is used and loss is a summation of localization loss and classification loss. Moreover, hard negative mining [44] and extreme data augmentation techniques are used to advance the detection accuracy. After SSD achieving state of art results against YOLO and Faster R-CNN [40], YOLOv2 [29] was proposed by Redmont et al. YOLOv2 has similar anchor mechanisms like SSD and outperforms the SSD with a more powerful backbone, batch normalization layers [45] and multi-scale training techniques.

2.5.4 Corner-Net

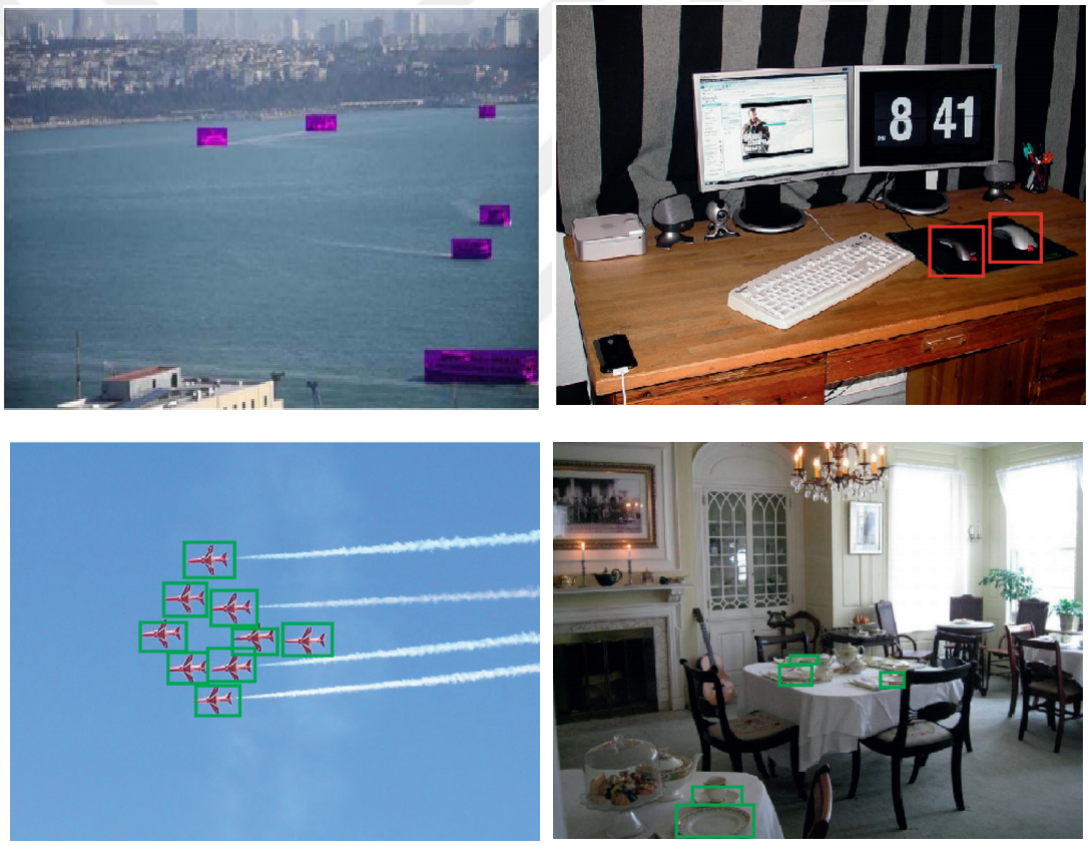
The one-stage detectors use prior information or regression techniques to localize and categorize objects without generating proposals. The first YOLO [43] detector approaches the localization problem as a regression problem. Then, SSD [5] proposed default boxes or anchor boxes which are derived from data set as prior information and have some hyper-parameters for involvement in the network. Also, YOLOv2 [29] followed the same way and brought the anchor boxes concept which is pre-computed before training with k-means clustering. Later, several studies have emerged on anchor-free detectors that predict localization information of objects without being associated with anchor anything but they tried to approximate the bounding box by predicting key points of them. Cornet-Net [46] was proposed as an anchor-free detector and pairs of corners are used to detect objects. The detection module of pair corners is responsible to calculate class heatmaps, pair embeddings, and corners offsets. The heat maps are used to detect corners as a group of same class objects. The top-left and bottom-right corners are extracted with a novel concept as a corner pooling which takes a maximum of row and column of pixels and gets a maximum point of intersection of column and row. The corner offsets are included in the regression problem and corner points of bounding boxes are refined. The Corner-Net achieves significant improvement among one-stage detectors on MS-COCO [33] data-set

Chapter 3

Small Object Detection

The object detection contains several challenges even though the modern object detectors achieve state of art results. One of the important challenges of object detection is detecting small objects. Small objects suffer insufficient strong visual information and bias of background or other larger objects make detection of the more difficult. A small object can be small in real world measurements or can be a normal object in the real world but it appears small in an image because of camera perspective Figure 3.0.1. The occlusion and overlapping between objects may increase the level of challenge in small object detection. The various object detectors have been proposed to advance object detection but the object detectors focus on the detection of objects that are categorized as medium or large, the small objects are still suffering in terms of low precision and recall. The feature maps of various detectors tend to miss the small objects because of their weak feature representation. The advance in object detectors proves that object detectors are becoming play a part in the real world and are widely used in various fields such as autonomous cars, authentications and security systems, medical imaging, robotics, etc. Then, these fields require the detection of small objects to achieve finding small faces, flinders, pedestrians, vehicles for military, surveillance, and robotic systems. This need occurs enhancement in small object studies and various studies have been proposed to deal with this challenge. In light of studies, we are going to examine approaches to overcome small object

challenges and introduce comprehensive comparisons and commentaries over the methods. In the next chapters, we will describe the definition of small object and introduce some of the preeminent datasets which are used commonly to train and evaluate the performance of object detectors and their formula of evaluation metrics. Then, we will represent approaches to overcome the small object detection challenge and compare the results. Finally, a comprehensive evaluation will be made.



(a) (b)

Figure 3.0.1. The objects(a) may have small appearance in the image on the contrary they can be considered as large, (b) objects appear small in the images as same as in the real life. The image in the first row of first column is taken from [3], the remained images are taken from [4].

3.1 The definitions of small objects

The definition of a small object is not a clear topic in the objection detection field. Visual representation of small objects can be described by both qualitative and quantitative properties. The qualitative description of small objects are hard to define in metrics because qualitative descriptions are relative. All objects can be negatively affected by occlusion, truncation, and overlapping with other objects or backgrounds in terms of object detection. The relativity occurs when these negatives factors effect the small objects more than other objects because of low pixel occupation. The low pixel occupation generate weak texture information when is combined with these negatives factors. The resulted weak texture may have insufficient sharpness, blurred regions, unrecognizable object appearance or indistinguishable pixels from another objects and backgrounds. Adding these characteristic of small objects to definition of them, they have to be measured for converting to metrics and determining generalized thresholds for further evaluation purposes. There are several measurements for intensity, sharpness, resolution, etc., but these metrics are not included to annotation of objects in datasets. Also, occlusion and truncation are taken into account for evaluation in some annotation of datasets but they are general properties for all objects. In short, there are some qualitative properties can be describe small objects but they are not specified in terms of measurements and this field is open for further research to contribute these properties in definition of small objects. On the other hand, the quantitative properties of small objects can have more clear metrics than qualitative properties and they are more about the size and scales of small objects that have to be defined in annotations of datasets. In spite of clearity in metrics, the quantitative properties have controversial side in terms of threshold because the size of input images are various and the ratio between object and image may differ. For example, if an object has a bounding box, has a size of 100x100, this bounding box is considered as a small object in a 2000x2000 image but for a 500x500 vice versa. The size ratio between object and image or directly size of image can specify the threshold but there is not a generalized definition for it yet. For instance, Zhu et al. [47] describe the small object as 20% of the size of an image. This definition uses size ratio between image and

object as threshold. On the other hand, Torralba et al. [48] mentioned another definition for small objects and they state that if an object has 32x32 or fewer pixels, it can be considered as a small object. This definition describes small objects in terms of their sizes and does not need to make correlation with image size. In the MS-COCO [33] dataset, they followed same definition of [48] and a small object is defined as the object that has equal or smaller than 32x32 pixels area. To sum up, quantitative properties can be used to make definitions of small objects more effectively and uncomplicatedly despite of difficulty to determine the threshold. In our study, we follow same convention that was used by previous studies commonly and we used definitions are derived from quantitative properties. Two different datasets are used to measure the accuracy of object detectors for small objects in our study. The MS-COCO and VisDrone-DET [8] datasets are used to train and test object detectors. Thus, we use two different small object criterion. For MS-COCO, we used same definition which is specified in the dataset and according to the dataset, every object have instance segmentation annotation, so if mask area of an object is smaller than 32x32 pixel area, it is considered as small object. On the other hand, we analyzed ratio between image and object size of VisDrone-DET and we do not specify threshold for the small objects because VisDrone-DET can be considered as custom dataset for small object detection. The VisDrone-DET mostly consists of small objects. The ratio of training objects that has bounding box area smaller than the 1% of the area of image is 97.5% and same ratio for validation objects is 97.7% for VisDrone-DET dataset. According to these statistics, we do not have to make distinct small object separation for VisDrone-DET dataset and all objects are considered as small objects in evaluation part.

3.2 The challenges in small object detection

The properties of small objects and the unsuitable architecture of object detectors causes challenges in small object detection. The reasons for the challenges can be grouped into four main topics: low pixel occupancy, weak feature representation, unoptimized parameters, and class imbalance.

The low pixel occupancy is one of the main drawbacks of small objects. The small objects compose the little part of pixels in an image, hence low spatial information decreases the learning capability of object detectors because of convolutional filters and pooling layers. The convolutional filters and pooling layers process the image to extract important parts of the image like a summarization. If building analogy between small object area and short sentences in a book, short sentences do not appear in summary as well as features of small objects can be already vanished before appearing in feature maps for detection. On the other hand, the architectural design of object detectors may cause an increase in this drawback. The small input size of object detectors makes the image resize to the lower resolution and degrade the pixel area of objects Furthermore, the usage of a single feature map in deeper networks may cause vanishing features of objects that occupy a small area in the image.

The weak feature representation is related to low pixel occupancy of small objects but the architectural design of networks is more responsible for emerging this drawback because many detectors are not designed for small objects. The relation between low pixel occupancy and weak feature representation addresses the disadvantage of usage of a single map once again. Besides that, backbone networks can be deep and cannot preserve the features of small objects against the enlarging in a receptive field which is caused by convolutional layers. The different layers contain different kinds of information about objects that have various scales, sizes, and aspect ratios. The features of these layers can be used to enhance the weak features with fusion techniques instead of straight flow in the process. The smaller receptive fields are descriptive for small objects so obtaining suitable receptive fields can solve weak feature representation.

The unoptimized parameters are directly related to the settings of object detectors. In the training phase, initializer weights of layers can affect the learning capability of object detectors. Also, hyperparameters of the network have to be suitable for training the dataset to improve training for better accuracy.

The distribution and characteristics datasets can be very challenging for object detectors and lead to inferior performance in detection especially for classes that

are hard to detect because of several drawbacks. The imbalance between classes or types of objects causes that ignoring some objects such as small objects. Balancing training data is one of the solutions for that problem. Also, giving more importance to small objects in the training phase with custom loss functions that can prevent bias of classes which are the majority and easy to detect.

The reasons for challenges in small object detection with their relevant solutions are shown in Figure 3.2.1 The associated approaches to solution in Figure 3.2.1 are explained in detail with their examples in Section 4.

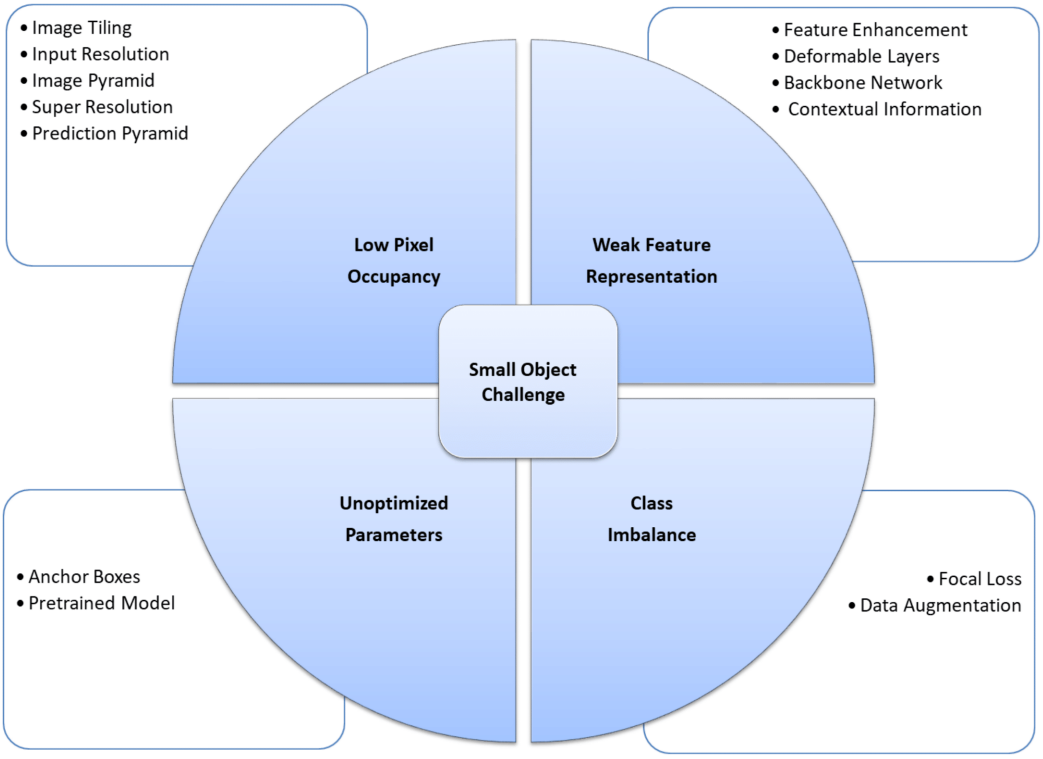


Figure 3.2.1: The reasons of challenges in small object detection and related solutions for them.

3.3 The datasets and evaluation metrics

The popular datasets which are used for benchmarking object detectors are commonly developed for general purposes such as segmentation, instance annotation, keypoints detection, etc. The general-purpose datasets contain various objects of different sizes. The most popular and challenging examples of the datasets are MS-COCO [33] and PASCAL VOC [49, 50]. Also, there are some datasets designed for the challenge of small object detection. These datasets aim to help for developing object detection networks that are customized for small objects. In this chapter, we will introduce both general-purpose datasets and custom datasets for small object detection.

3.4 The general evaluation terms

3.4.1 Intersection over union

The classic object detectors predict class and bounding box information of objects. The predicted bounding box of an object has to meet the criterion for the ground truth bounding box of the same object during training. The criterion concept consists of a measure and threshold to determine the success of the training process. Intersection Over Union (IOU) uses the Jaccard Index as a measurement and computes the IOU between ground truth and predicted bounding boxes. The IOU is computed as an intersection area divided by union area of ground truth and predicted bounding boxes. Then, IOU is compared with the threshold to determine whether the detection is correct or wrong. In other terms, true positive(correct detection) or false positive(wrong detection) are determined. The mathematical expression of IOU shown below in the equation (3.1).

$$IoU(A, B) = \frac{A \cap B}{A \cup B} \tag{3.1}$$

3.4.2 Precision and recall

Precision is the measure of the correctness of predictions. In other terms, the ratio of true positive samples overall detections is the sum of true positive and false positives (3.2).

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{all\ detections} \tag{3.2}$$

Recall measures the detection ability of an object detector with only counting correct detections(true positives) and compares with all ground truths. The true positive detections divided by all ground truths gives recall performance (3.3).

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{all\ ground\ truths} \tag{3.3}$$

3.4.3 Precision-recall curve

The IOU, precision, and recall are important metrics to determine some part of object detectors' performance but they are not enough when they are alone to evaluate the overall performance of the detectors. Thus, the precision-recall curve is used to evaluate performance and detect bottlenecks of detectors. The performed-well detector draws a curve that precision increases with recall remain high or vice versa. Moreover, if the count of false positives and false negatives are close or equal to zero, we can consider that the object detector is fine to detect all ground truths with correct identification.

3.4.4 Mean average precision

The average precision is derived from the area under the precision-recall curve. The idea behind average precision is that presenting a scalar metric which helps

to compare different detectors easily because the average precision curves do not follow specific patterns and tend to generate zigzag plots. The mathematically average precision is the averaged all precision values between recall intervals as 0 and 1. The average precision is calculated per category or class, if we want to obtain mean average precision, we simply average the overall precision of categories. Furthermore, the recall intervals may be chosen differently to calculate the area under the curve. The PASCAL VOC [49, 50] and MS-COCO [33] challenges have different ways to calculate Mean Average Precision (mAP).

3.4.4.1 PASCAL VOC method

Before 2010, PASCAL VOC [49, 50] challenge used the 11-point interpolation calculation. The 11-point interpolation divides recall levels between 0 and 1 to equally 11 points as [0,0.1,0.2...,1]. The interpolation function takes each point and gets a maximum precision score between the current point and endpoint (1.0). Then, all maximums of 11 points are averaged to obtain average precision. After 2010, all points interpolation calculation is used to get average precision. The all-points interpolation takes maximum precision points on the precision-recall curve and calculates the area between these points to average them. The final mean average precision is obtained from the average of calculated average precisions. In PASCAL VOC, the IOU threshold is taken as IOU>0.5 and all levels of IOU have equal contribution to the calculation of precision. The mathematical expression of calculation of average precision from 11-point interpolation is shown below in the equation (3.4).

$$AP = \frac{1}{11} \sum_{r \in (0,0.1,\ldots,1)} \max_{\tilde{r}:\tilde{r} \geq r} p(r) \tag{3.4}$$

3.4.5 Precision-Recall Curve

3.4.5.1 MS-COCO method

MS-COCO [33] method differs from PASCAL VOC [49, 50] in terms of IOU thresholding and sampling of recall points. It uses a range of IOU for thresholding that ranges with a step size of 0.05 from 0.5 to 0.95. Also, recall sampling points are arranged to 101 points represented as [0:.01:1]. If the threshold is specified as single points, it is represented as AP50 or AP75 that means the IOU threshold is 0.5 and 0.75 respectively. The mAP at each IOU point in the range [.5:.05:.95] is calculated and averaged to obtain the final mAP (3.5).

$$mAP_{COCO} = \frac{mAP_{0.50} + mAP_{0.55} + ... + mAP_{0.95}}{10} \tag{3.5}$$

3.5 The general purpose datasets

3.5.1 PASCAL VOC

The PASCAL VOC datasets are divided into two main versions as VOC2007 [49] and VOC 2012 [50]. Each version of the dataset could be considered as a mid-level dataset and contains objects from 20 different classes. Both versions are annotated for evaluation of several tasks in computer vision fields: object and action detection, segmentation. The dataset splits consist of training, validation, and test. The VOC2007 contains 2501, 2510, 5011 and VOC2012 contains 5717, 5823, 10991 images in order of training,validation and test.

3.5.2 MS-COCO

MS-COCO [33] has become one of the most famous and challenging datasets since it was published in 2015. It has 80 classes and 897k annotated objects from 164k images (MS-COCO-2017). Each object has an instance segmentation. The MS-COCO-2017 splits into three sub-datasets: training (118,287), validation (5000) and test (40,670) images. We used MS-COCO in our comparison section for techniques that will be explained in further sections because MS-COCO is used for benchmarking nearly all papers and contains a considerable amount of small objects in the dataset. Furthermore, the criterion of being considered a small object is determined and accepted as objects have instance masks smaller than 32x32 pixels. The small objects present 41.43% of all objects and appear 51.82% of all images in the MS-COCO-2017 dataset. However, if we look at the pixel occupation of small objects, it becomes challenging because small objects occupy only 1.23% area of images pixels.

3.6 The small objects datasets

3.6.1 TinyPerson

TinyPerson [51] was introduced as a challenge for detecting people in images where they are far away from the camera and occupy little portions of images that have huge backgrounds. The new terms which are absolute and relative size of objects are used for benchmarking. The dataset aims to help detection of people that present near the sea for quick maritime rescue and defense around the sea. The images contain over 200 persons densely. The average precision and miss rate are used for performance evaluation. The objects are divided into 3 sizes: tiny (2,20) ,small (20,32) and all (2,inf). Furthermore, tiny objects are divided into three sub-sizes : (2,8), (13,20) and (13,20). 0.5 and 0.25 are used as the IOU threshold for performance evaluation. The IOU criterion is modified for the ignored regions in the dataset and named as Insertion of Detection (IOD).

3.6.2 DOTA

Dataset for Object Detection Aerial Images (DOTA) [52] is introduced to enhance object detection for remote sensing, satellite or aerial imaging. It has 14 main classes of objects in different orientations in the 2806 images which vary their sizes between 800x800 to 4000x4000. The total number of instances is 188,282 and 67.10 instances are annotated for each image averagely. The aspect ratio and size of objects have huge variation and extreme objects can be found such as bridges which have extreme aspect ratio. The extreme differences between objects make the dataset more challenging. In terms of small objects, aerial images contain huge amounts of small objects with massive backgrounds. The DOTA contains objects which have horizontal bounding boxes [53] smaller than 300 pixels in a ratio of 98% and nearly half of small objects have less than 50-pixel horizontal bounding box.

3.6.3 VisDrone-DET

The VisDrone [8] dataset contains several challenges for computer vision fields, that are image object detection (DET) , video object detection (VID) , Single Object Tracking (SOT) and Multiple Object Tracking (MOT). In this survey, the image object detection part is considered for investigating small object detection. The dataset is a collection of images and videos from drones and contains objects in different sizes and aspect ratios densely. VisDrone-DET contains 10,209 images which are annotated for 10 categories. The dataset is divided into training (6,471), validation (548), test-challenge (1,580) and test-dev (1,610).

3.6.4 WIDER FACE

The WIDER FACE [53] is developed as a dataset to advance face detection challenges with 393,703 annotated faces in 32,203 images. The dataset have a huge variation in terms of facial expression, occlusion, size, illumination, etc.

Chapter 4

The Approaches for Small Object Detection Challenge

In recent years, many kinds of research have been conducted on object detection networks and some of them put effort into increasing the performance of small object detection. We grouped the proposed ideas into three main categories: pre-post processing techniques, network architecture solutions or modifications, and regularizing hyperparameters or priors of the network. In this research, the evaluation and comparison of approaches into four main groups which we introduce for fairness: intra-comparison, intercomparison, global comparison, and a group of exclusion from the comparison. The intra-comparisons are done for mostly pre-post processing and regularizing hyperparameters or priors on object detection networks. In this type of comparison base network which produces results are considered as base scores for benchmarking, and a regulated base network has almost identical network architecture. The intercomparison is used for network modifications such as Deconvolutional Single Shot Detector (DSSD) [54] and Feature Fusion Single Shot Detector (FSSD) [55] which are derived from SSD [5] network. In global comparison, some of the approaches may not be compared directly with other networks because they may be not related to other networks in terms of architecture so we compared them in a global comparison table. Finally, some approaches are dedicated to very specific parts of small

object detection such as tiny face detection, the contributions of these approaches are valuable for the detection of small object challenge but we cannot directly compare them with other works.

4.1 The pre and post processing techniques

4.1.1 Data augmentation

Data augmentation is one of the effective techniques to improve training accuracy. The capability of the generalization object detection model is increased by augmenting data by applying different image processing transforms on training data. The over-fitting, most undesirable problem in training object detection networks, can be overcome with data augmentation. The over-fitting means that the network generalizes training data well because of the curse of dimension but produces bad predictions on validation or test data. The most common reasons for over-fitting are lack of training data to fit the network and using very deep networks for training. The data augmentation deals with lack of training data intend that more information can be gathered from the original dataset through augmentations such as data warping or oversampling [56]. The data augmentation includes applying transforms such as geometric transformation, color-based transformation, mixing images, frequency domain transformations, and deep learning approaches. The result of data augmentation is increasing training data and trying to increase the learning capability of networks with augmented features.

The data augmentation becomes a crucial method for state of art object detection networks. For example , SSD [5] networks benefit from data augmentation and increased average precision on PASCAL VOC [49, 50] and MS-COCO [33] datasets. Figure 4.1.1 shown below shows the amount of improvement in the mAP for the VOC 2007 test with or without data augmentation. According to the study, SSD achieved +8.8% mAP from baseline with their augmentation strategy on the VOC 2007 test set.

	SSD300				
more data augmentation?		✔	✔	✔	✔
include $\{\frac{1}{2}, 2\}$ box?	✔		✔	✔	✔
include $\{\frac{1}{3}, 3\}$ box?	✔			✔	✔
use atrous?	✔	✔	✔		✔
VOC2007 `test` mAP	65.5	71.6	73.7	74.2	**74.3**

Figure 4.1.1: The figure shows the effect of data augmentation for testing accuracy with other anchor and convolution layer settings [5].

If we focus on improvement for small object detection, there some dedicated data augmentation techniques have been proposed. Kisantal et al. [6] studied several data augmentation methods for small objects. Their augmentation method is based on two different strategies: oversampling images that contain small objects and copy-paste small objects to images synthetically with different ratios. They used the MS-COCO dataset and Mask R-CNN network. In their evaluations on different pasting augmentation strategies, they achieved 9.7% relative improvement on the instance segmentation and 7.1% on the object detection of small objects. Their main goals are increasing the representation of small objects in training data and matching anchors with small objects in training. They stated that the best combination for data augmentation is that using the original images with small objects and their copy with copy-pasted small objects are used for training. The difference between baseline and proposed is shown in Figure 4.1.2 below in terms of average precision.

	Segmentation AP				Detection AP			
	small	medium	large	all	small	medium	large	all
baseline	0.113	0.300	0.418	0.28	0.167	0.329	0.393	0.303
aug	0.108	0.299	0.422	0.278	0.161	0.328	0.4	0.302
aug+oversample 2x	0.117	0.300	0.406	0.277	0.168	0.333	0.387	0.302
original+aug	**0.124**	0.301	0.41	**0.28**	**0.179**	0.329	0.386	**0.304**

Figure 4.1.2: The figure shows the effect of proposed data augmentation on average precision of small objects [6].

As a result, the main advantage of data augmentation achieves improvement for small object detection accuracy without changing the network. On the other hand, the method of data augmentation produces two drawbacks: computation time which is increased by online augmentation because of image processing in the training phase, or amount of storage which is increased by offline augmentation because of producing new training data. These drawbacks can be negligible beside the improving accuracy.

4.1.2 Input size of network and image tiling

The input sizes of object detection networks vary with the objective of the network. The sacrifice between performance and precision determines the input size of the network because the size of the whole network is shaped by the input size. When the input size of a network, memory consumption and inference time increases. Therefore, the network is getting to fall down on real-time requirements and memory constraints, this situation is not suitable for embedded systems. On the other hand, the precision of the network can be higher and predictions become more accurate because high-level feature maps represent more details of objects from data under the assumption of overfitting does not occur in training. The large input sizes also affect the accuracy of small object detection positively because the convolution layers and max-pooling layers remove the weak features which are mostly small objects.For example, small object,image and network have input sizes respectively; 32x32, 1024x1024 and 256x256. For this network, the image has to be resized with a factor of 0.25 to fit the network. So, the size of a small object becomes 8x8 with the same ratio. Moreover, if we apply a pooling operation to this object more than 3 times, the feature representation of the object becomes under 1 pixel and we consider that the object vanishes for prediction. layers [4].

The image tilling means that instead of resizing the image to network size, the image is divided into overlap or non-overlap patches, and each patch is fed to the network. The image can preserve the original quality and details of features

concerning the ratio between the patch and network size. Therefore, the image cannot be affected by high ratio scaling and small objects preserve their sizes. On the other hand, the image tilling can increase the inference time and bring extra post-processing overhead for merging and refining predictions. In this research [7], PeleeNet [57] is used with image tiling for detecting objects in the VisDrone [8] dataset which consists of pictures that are taken from drones. The process flow diagram of the method is shown in Figure 4.1.3.

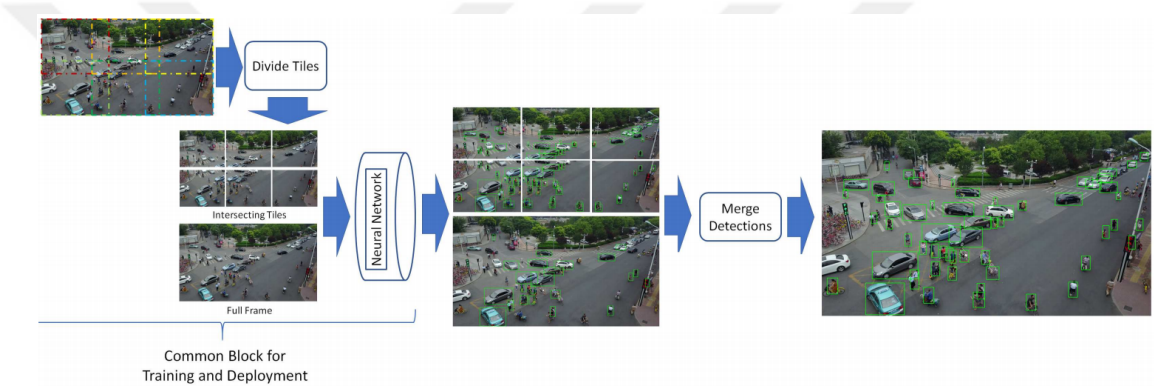


Figure 4.1.3: The proposed tiling process flow diagram for training and testing [7].

The different sizes of image tiling grids have been experimented with for both training and evaluation phases. Also, the real-time performance of image tiling configurations was considered for embedded systems platforms such as Nvidia Tx1 and Tx2. The authors tried to accomplish better accuracy and admissible frame of rate with a lightweight detector and image tiling. They used two configurations of PeleeNet: the modified version which used the 38x38 feature map and the default version. The images are resized to 1920x1080 for equal patch sizes and tilled with 25% overlap for keeping continuity of objects at boundaries. The original resized image and patches of image fed into the network for training and testing phases. In the test phase, the detections of both the original image and patches are merged with the proposed merging technique. The criterion of merging was stated by authors that if the intersection of duplicate detections is above 25%, then the one with a higher score is accepted as a better choice and the other one is removed from the detection list. The results of best tiling setups are

compared with their baselines in Figure 4.1.4 and the evaluation results which changes according to different tiling setting in both training testing shown at Figure 4.1.5.

Model	Avg. Precision (%), Iou:									FPS on:	
	Vehicle			Pedestrian			MAP				
	0.5	0.75	0.5:1.0	0.5	0.75	0.5:1.0	0.5	0.75	0.5:0.95	TX1	TX2
Vino	**32.27**	17.29	17.21	**24.74**	3.64	8.75	**28.50**	10.46	12.98	11.7	16.5
Pelee	17.03	5.68	7.39	5.93	0.18	1.43	11.48	2.93	4.41	58	101
Pelee38	28.45	14.30	14.95	17.97	1.28	5.28	23.21	7.79	10.11	48.2	77.8
Vino_I2x1	33.30	15.40	17.17	24.51	3.60	8.75	28.91	9.50	12.96	5.7	8.3
Pelee_T5x3_I5x3	41.39	19.55	21.07	**30.26**	2.94	9.61	35.82	11.24	15.34	3.5	6.3
Pelee38_T5x3_I5x3	**44.35**	22.64	23.53	28.99	3.25	9.69	**36.67**	12.95	16.61	3	4.8

Figure 4.1.4: The results of best tiling settings for different version of Pelee Network and Vino on Nvidia Tx1 and Tx2 with VisDrone [8] dataset [7].

T/I	3x2	4x4	5x3	6x3
Single	26.55	25.77	28.48	27.81
3x2	30.63	34.62	34.11	31.69
5x3	28.57	30.86	35.82	35.88

Figure 4.1.5: The changes in average precision when tiling settings are changed in both inference and training stage [7].

4.1.3 Pretrained model

The accuracy of the object detection model is affected by the training dataset. More and better data increases the generalization capability of a model. However, training a model with specific data from scratch is very difficult because the generation of an adequate specific dataset is costly. The insufficient dataset leads to overfitting of models. The pre-trained models prevent this problem and make it easier to train specific datasets. The general convention for this technique is that an extra dataset that is more general and bigger than a specific dataset is trained on a model. Then, this pre-trained model is fine-tuned with the specific dataset to achieve the desired task [51]. The extra dataset improves the generalization

capability of the model and leads to the enhanced performance of the fine-tuned model. This approach can be used for dealing with small object challenges in terms of imbalance class problems and data sufficiency. However, according to research [51], this improvement is limited because extra data and main data are very different in terms of the distribution of scales and sizes of objects. The authors of the research proposed a scale matching technique to achieve more improvement from pre-trained based detectors for detecting small objects. The scale matching uses an analogy of histogram matching between two images and it generates a derived dataset that has the same distribution of scales and sizes of objects as the main dataset which has mostly small objects. Thus, the new pre-trained network which is trained by the derived dataset can have more information about small objects and the result of fine-tuning on the main dataset is boosted. In the research, MS-COCO [33] was used as an extra dataset and was processed with proposed scale matching to resemble the distribution of TinyPerson dataset in terms of scales and sizes of objects. The Faster R-CNN-FPN [41] network is pre-trained with a scaled MS-COCO dataset then fine-tuned with the TinyPerson dataset. As a result of this experiment, the performance of Faster R-CNN-FPN was improved 5% over the baseline.

4.2 Network architecture solutions

4.2.1 Backbone networks

The backbone networks are the core of the object detectors. They extract features from input images and generate feature maps which are high-level representations of images. The backbone networks can affect the accuracy of small object detection. The number of layers, kernel sizes of convolution filters, the flow of features between layers, normalization layers are the keys to performance and accuracy. However, more deeper or complex backbones do not always lead to top accuracy because insufficient training data or low visual representation of objects like small objects cannot give enough information to the network for updating their large

parameters and thus over-fitting occurs. Also, the complexity of networks does not mean low inference speed, the inference speed depends on the capability of hardware utilization. The backbone networks have to be chosen carefully with respect to the characteristics of the training dataset. The features of small objects can vanish in deeper networks because every feature extraction step tends to extract the strongest features in images and small objects are staying out of the learning process. The residual blocks, shortcut connections, and different sizes of convolution kernels can enhance the accuracy of small object detection.

4.2.2 Multi-scale detection

One of the important challenges in the object detection field is developing scale and aspect ratio invariant object detectors. Multi-scale detection has been come into use as a conception to overcome this challenge. Also, this concept has proved itself for dealing with the detection of small objects within years. The researchers have begun with sliding windows techniques on feature pyramids, especially traditional methods such as HOG [14], DPM [21] and Overfeat detector [42] used this technique to accomplish multi-scale detection. The more complex dataset and appearance of objects bring new ideas to the multi-scale detection field. The mixture models and exemplar-based detection techniques were developed to deal with complex datasets before the beginning of the deep learning-based object detectors. At the beginning of deep learning-based object detectors, object proposals were developed for deciding the regions of interest which areas have high potential to contain objects in an image. The object proposals deliver low computation time with a high recall rate and localization accuracy than ancestor techniques. However, the generation of object proposals as an extra step, the evolution of GPUs, and the need for real-time object detectors pushed the researchers to search for different techniques such as deep regression and multi-reference or resolution detection techniques. The deep regression technique aims to predict the bounding boxes of objects by making regression on extracted features. The YOLO [43] detector, the pioneer of one stage detectors, uses a deep regression method. However, this method suffers from inaccurate localization

especially for small objects [10]. The most advanced and recent approach for multi-scale detection is multi-scale feature learning which is used from prominent object detectors. The idea behind multi-scale feature learning is combining multi-reference and multi-resolution techniques to gain power in both reference/anchor boxes and different resolutions of feature maps. The anchor or reference boxes which are explained in 4.3, is the important concept of multi-reference detection. The optimization methods and importance of anchor/reference boxes are out of scope for this section, they will be introduced in the next sections. The idea behind the multi-scale feature learning can be explained with the understanding of both feature maps and input sizes in CNNs. We examined high-resolution inputs that are more suitable for small object detection because they can contain more spatial-rich information in deeper layers. Multi-scale learning aims to gain power for detecting more objects by using different resolutions of input images and feature maps in the network. The single feature map can be insufficient because layers of the network become deeper, relative resolution becomes more coarse than input size and convolutional filters remain bigger receptive fields which are more suitable for large objects. Thus, the small objects are more detectable with high-resolution feature maps with smaller receptive fields in shallow layers of the network.

Multi-scale detection can be divided into several main process flows and the numbers and naming of paradigms may change because of the interpretation of authors and they are not split crisply. For example, according to Wu et al. [9], multi-scale feature learning can be divided into four main paradigms: Image Pyramid, Feature Pyramid, Integrated Features, and Prediction Pyramid. In our study, we categorize the multi-scale detection into two main paradigms: Image Pyramid and Feature Pyramid and we divide feature pyramid paradigm to prediction pyramid and integrated prediction pyramid are special sub paradigms of feature pyramid. The diagrams of image pyramid and types of feature pyramid are shown below in Figure 4.2.1 and Figure 4.2.2.

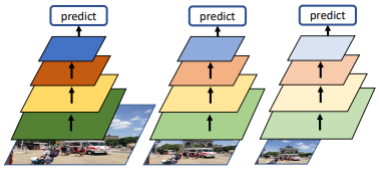


Figure 4.2.1: The diagram of image pyramid [9].

(a) (b)

Figure 4.2.2. The types of feature pyramid: (a) Prediction pyramid, (b) Integrated prediction pyramid [9].

4.2.2.1 Image pyramid

The simple CNN based object detector makes predictions on a single feature map. The image pyramid aims to train multiple detectors with different input sizes and these detectors produce different sizes of feature maps. Then, predictions on feature maps are combined to obtain a refined prediction of the image. The image pyramid advances the detection of small objects by producing high-resolution feature maps from high-resolution images of the image pyramid. However, training and inference of multiple detectors increase computation time and memory usage. Image pyramid usually does not meet real-time requirements for utilization of hardware in terms of parallelism and memory. There is significant research on the image pyramid technique to improve object detection accuracy.

Liu et al. [58] proposed a face detection network that uses current scale approximation and scale-forecast network modules to build scale aware network. The scale-forecast network was used for determining an appropriate scaling factor for the image pyramid. Current scale approximation was used for learning

feature maps on different scales in the manner of smaller ones from larger ones. in light of scale-forecast network.

Singh et al. [59] investigated the small object detection problem extensively and proposed an object detector which was called Scale Normalization for Image Pyramids (SNIP) based on the image pyramid paradigm. They stated that feature pyramids or integrated features are not the best solutions for improving the detection of small objects. The SNIP uses image pyramids with region proposal network and recursive cortical networks to refine region proposals at each scale of the image and predict correspondence objects for each scale. The different object detectors were trained for each scale of the image to handle scale invariance and improve small object detection. They use the sub imaging technique to avoid using too much memory for high-resolution images. Furthermore, this research was extended with Scale Normalization for Image Pyramids Efficient Resampling (SNIPER) [60] to reduce training time and memory consumption and benefit from larger batch sizes. The SNIPER extends the SNIP with efficient resampling and proposes a new sub imaging technique. The new sub imaging technique was referred to as chip mining and divided into two parts as negative and positive chip mining. The positive chip mining was used for generating sub-images that contain ground truth boxes. On the other hand, negative chip mining prevents false positive detections that came out from background parts of the image. The region proposal network is used for generating negative chips before the training then they are used in the training phase for speeding up learning by processing fewer pixels. Also, the larger batch sizes which are the result of chip mining and batch normalization technique increase the accuracy of the detector.

Meng et al. [61] researched the detection of small traffic signs from large images and proposed an object detector, Small Object Sensitive (SOS)-CNN, which is based on an SSD [5] using an image pyramid. The SOS-CNN splits the images at each scale into patches and patches are fed to the network. The detections which came from each patch are merged by non-maximum suppression. Multi-patch detection preserves real-time inference speed because large images split into fixed-size batches and batch inference is used for process patches. The top-most feature map is used for prediction because the network is designed for small

objects and the top-most feature map has spatial-rich information to reveal small objects but the detection performance of larger objects is reduced. The larger objects are handled as small objects with help of small scales of the original image by image pyramid.

Another SSD [5] based network was proposed by Pang et al. [62] to exploit the effectiveness of the image pyramid paradigm. Their design uses different scales of an original image to enrich the multiple feature maps in the SSD network with a special fusion module. Each level of the pyramid fed into a convolutional block to extract features and feature attention modules are used to combine these features with corresponding feature maps that come from SSD layers. Also, each combined feature map is fused again with a forward consecutive feature map. This network also can be considered as an example of feature aggregation.

IPG-Net [63] which is based on FPN [41], is a specialized network for small object detection. The proposed network combines the FPN structure with the image pyramid paradigm. The network consists of special IPG transformation and fusion modules instead of identical branches for each image in the pyramid. The spatial information of shallow layers is transferred precisely without loss to deeper layers and provide enough spatial information for small objects to be detected in prediction layers. The semantic-rich information is combined with spatial information at each level of the image pyramid. IPG-Net achieved good performance to eliminate an imbalance of spatial and semantic information for feature maps.

4.2.2.2 Prediction pyramid

Instead of using multiple detectors with different input resolutions, the prediction pyramid gathers features from shallow and deeper layers of the network to produce multiple feature maps from a single network. The predictions are made on hierarchical feature maps. The feature maps have different resolutions and receptive fields to detect distinct objects with different scales. Therefore, detection of small objects is improved with the feature maps which are produced from

shallow layers of the network.

SSD [5] is a famous object detector that uses a prediction pyramid system. The detector combines spatial and semantic features from different points of the network and makes predictions on multiple feature maps at different scales. SSD variant detectors are similar but feature aggregation techniques change the paradigm of the detector to an integrated prediction pyramid with fusion modules or neck parts.

YOLOV3 [30] is the first user of the prediction pyramid system in the YOLO family. It makes predictions with YOLO layers at three different scales of feature maps. It uses a well-designed new backbone as Darknet-53 which is deeper and includes more shortcut connections. The larger networks facilitate the collecting semantic-information and shortcut connections prevent the vanishing of gradients of shallow features in deeper layers. The YOLOV3 has a significant accuracy difference in small object detection performance than previous YOLO detectors [43, 29] which use a single scale feature map for detection.

TridentNet [64] is the scale-aware object detector that is based on multi-branch object detection. TridentNet combines image pyramid with prediction pyramid paradigm but rather than using multi-input with single or multi-branch networks, it uses single input images with parallel branches. The parallel branches are architecturally identical but using dilated convolution within a range of stride makes difference. Each branch produces feature maps in different scales and receptive fields with help of dilated convolutions. Furthermore, parallel branches share weights between them to make training easier and stronger. The network can learn which objects more suitable to one of the hierarchical branches according to their scales.

4.2.2.3 Integrated prediction pyramid

The integrated features technique is similar to prediction pyramids but it differs in a manner of collection of feature maps. The feature maps from multiple layers are fused into a single feature map or share features to construct multi-scale feature maps and predictions are made on the combinatorial feature maps. Shallow layers of the network have rich spatial information and deep layers contain semantic-rich features thus, the combination of shallow and deep features boosts the object detection at different scales because integrated features map can give more information about objects and improve detection accuracy and recall [9]. The small objects vanish in deeper layers because of insufficient spatial information and large receptive fields. The shallow features in the integrated feature map protect the spatial information of small objects with small receptive fields. The networks that use the integrated features paradigm will be described in section Feature Aggregation and Enrichment comprehensively.

4.2.3 Feature aggregation and enrichment

The recent advances in deep learning-based object detectors develop more complex modules for exploiting the quality of extracted features. In the multi-scale feature learning, integrated features and feature pyramid approaches aim to fuse features from different feature maps. The fusion modules between layers or maps can be simple connections like in the FPN [41] modules or they can be more complex for improving feature aggregation and enrichment. These modules, named the neck, are usually inserted between backbone and prediction or head module of the network [1]. The neck modules can be extra modules that extract and/or fuse features from different layers of the network to enrich feature maps or can be path aggregation modules that connect and fuse feature maps. The path aggregation modules usually are used in FPN-like networks in the manner of bottom-up and top-down connected layers to generate final feature maps. The extra modules can be simple fusion blocks such as deconvolution layers with concatenating layers or more complex modules such as Multi Scale Contextual

Features (MSCF) with Feature Guided Refinement Module (FGRM) in EFGR-Net [65] network. The neck modules can increase the computation time but a suitable fusing design improves the accuracy of object detection. Furthermore, the feature enrichment is achieved without adding extra modules to the network by modifying the characteristic of existing layers or developing new layers to collect more features or preserve them. The dilated max pooling and convolution layers are some examples of this approach. In the perspective of small object detection, feature aggregation and enrichment are very important concepts to improve detection because small objects suffer a lack of feature representation and artifact of growing receptive fields in deeper layers of the network. These modules and modifications behave like amplifiers, feedbacks, or more sensitive filters in the processing flow of the network. They amplify the features of small objects with upsampling layers. They behave like feedback with skip connection and fusing techniques and bring features that are lost in the flow, to final feature maps.

4.2.3.1 SSD variants

This section explains networks that are derived from SSD [5] and their feature aggregation or enrichment techniques.

DSSD [54] extends the SSD [5] network with deconvolutional blocks. Instead of VGG-16 [26] which is the default backbone of SSD, DSSD uses ResNet-101 [28]. The deconvolutional blocks increase the resolution of the feature map and their features are supported with skip connections from SSD layers. The features at different scales are combined and amplified with deconvolutional blocks. This feature enhancement technique performs better for small object detection than SSD as a baseline but deconvolutional blocks bring extra computation and decrease frame per second.

Jeong et al. [66] proposed a feature fusion module between SSD [5] layers. The proposed network aims to increase the number of channels in SSD layers for generating better feature maps. The features from shallow and deeper layers are

combined with rainbow fusion which is the naming of concatenation that consists of both pooling and deconvolution layers. In the aspect of small objects, the proposed network unifies the features between feature maps at each scale and prevents imbalance in features especially for small objects.

Cao et al. [67] designed a feature fusion module for SSD [5] layers to increase the accuracy of small object detection without losing inference speed too much. The new network is named as Feature-Fused SSD. The proposed feature fusion combines a shallow conv4_3 layer with conv5_3 as a deeper one. This fusion enhanced the feature map which has the highest resolution in SSD in terms of both semantic and spatial information. The lightweight fusion module has two different types in terms of the feature fusion layer. These are concatenation and element-sum modules. The element-sum module slightly performs better than concatenation. The accuracy of Feature-Fused SSD comparable to DSSD [54] for small object detection and inference speed outperforms the DSSD.

FSSD [55] advanced the SSD [5] network with a lightweight feature fusion module for being more sensitive to small objects. The idea behind the fusion module is that collecting semantic information from deeper layers and creating contextual information for small objects for layer conv4_3 of SSD. It aggregates features of feature_map (con4_3) with subsequent layers as fc7 and conv7_2. The accuracy of FSSD outperforms DSSD [54] andSSD for small objects detection but it falls behind in terms of inference speed when compared with SSD as the baseline.

Zhao et al. [68] proposed Comprehensive Feature Enhancement (CFE) module for SSD [5] network to improve object detection accuracy. They designed two different networks with different orientations of CFE modules in the SSD network. CFE-SSDv1 and CFE-SSDv2 are designed to enrich semantic information of shallow layers for better classification of small objects and to increase the accuracy of bounding box regression for deeper feature maps. The architecture of the CFE module is similar to the ResNeXt [31] building block. It contains residual skip connection, group convolutions, and concatenation layer. The proposed networks perform better than SSD as the baseline in terms of object detection accuracy

especially for small objects with a few amount of increase in inference time.

The effectiveness of feature pyramid on objection detection has been proved with FPN [41] variants object detectors. Zhao et al. [69] proposed a network based on SSD [5] with the inspiration of the effectiveness of feature pyramid. They named their work as a Multi-Level Feature Pyramid Network/M2Det. This network combines SSD architecture with Thinned U-shape Feature Aggregation Module (TUM)s. Also, the feature fusion modules are used to enrich feature maps from both SSD layers and TUMs. The base features, extracted from the backbone, are equally scaled and concatenated into a channel with Feature Fusion Module (FFM)v1, then they are fed into TUMs to enrich semantic information of feature maps at each scale. Therefore, another fusion block name as FFMv2 which helps to share features between TUM modules. The generated multi-scale features are aggregated with the Scale-wise Feature Aggregated Module (SFAM) to generate final feature maps for prediction. The TUMs make the network deeper to gather more semantic and representative features. On the other hand, the disadvantage of a deeper network is elevated with feature fusion modules across the feature maps and layers from different scales. The feature maps can be more distinctive for objects with different scales. Thus, the accuracy of small object detection is enhanced.

EFGRNet [65] was introduced as a significant improvement on SSD [5]. The design of the network aims to solve a lack of contextual information and foreground-background class imbalance in SSD. The MSCF module was designed to gather multi-scale contextual features from downsampled input image whose resolution matches with conv4_3. These features are used to enrich feature maps at each scale and refine bounding box and classification results with help of FGRM. MSCF module works like multi-branch modules in ResNeXT architecture with dilated group convolutions. MSCF modules at each level of the feature map are cascaded from the output MSCF module so contextual information passes through all feature maps. The small objects suffer a lack of foreground-background discrimination because of their low visual information and inappropriate anchor boxes. The proposed network enhanced the discrimination ability of SSD between objects and background. Thus, small objection detection performance increases

significantly.

4.2.3.2 YOLO variants

This section explains networks that are derived from different versions of YOLO and their feature aggregation or enrichment techniques.

Du et al. [70] enhanced the YOLOv2 [29] network and proposed a new network as Expanding Receptive Field (ERF)-YOLO which uses deconvolutional and multi-branch dilated convolution to improve both semantic and spatial information gathered from different layers at depth. The spatial rich information from the shallow layer is processed by the ERF block to increase receptive fields and enrich semantic features of the feature map. On the other hand, semantic-rich information from the deeper layer is upsampled with a deconvolution block to gain spatial information for the feature map. Then, two feature maps are fused into a single feature map for making predictions. These transformations aim to provide precise location information for small objects by increasing semantic information and boost the weak feature representation of small objects by upsampling.

Unmanned Aerial Vehicle (UAV)-YOLO [71] was proposed to solve the low performance of small object detection in UAV imaging. The network aims to increase receptive fields of feature maps and gather semantic information for feature maps which are generated from shallow layers. The residual blocks in Darknet-53 [30] backbone are increased with new residual blocks that have more skip connections from previous layers to improve the receptive field of the network. The semantic-rich feature maps are upsampled and concatenated with consecutive shallow maps to increase descriptive features in spatial-rich feature maps. The increase in receptive areas and descriptive features enhance the detection accuracy of small objects when comparing YOLOv3 which is a based network for UAV-YOLO.

YOLOv4 [1] was designed on top of the YOLOv3 [30] detector with significant modifications on backbone and connection between the backbone and detector

part which we called the neck in previous sections. The backbone that is named CSPDarknet53 is a modified version of Darknet53. The CSPDarknet53 was inspired by the novel skip connection technique in CSPNet [72] backbone. Besides backbone modification, SPP [38] is used to increase receptive fields in the network and Path Aggregation Network (PAN) [73] is used for gathering multi-scale information from different layers of the network to achieve feature aggregation. The SPP module provides a group of pooling operations on feature maps. This pooling operation contains pooling layers with different strides to produce fixed-length feature vectors that represent objects at different scales. Thus, receptive fields of feature maps are increased and detection cannot be affected by the variation of scale. The PAN is used for sharing features between feature maps at each scale. Therefore, shallow layers benefit semantic-rich information from deeper layers and deeper layers are fed with spatial-rich information from shallow ones. These modifications redound small objection detection performance in terms of more sensitive receptive fields and descriptive feature maps.

4.2.3.3 FPN based detectors

In the prediction pyramid technique, the feature maps do not share any features between them. The feature pyramid networks take the prediction pyramid technique and combine it with an integrated feature style. The multiple feature maps share features between them and improve their spatial and semantic information. The predictions are the same as the prediction pyramid but the difference is every feature map contains information from different feature maps at different scales. The feature pyramid approach improves small object detection with multiple feature maps that have rich spatial or semantic information at different scales. The first example of an object detector that is based on a feature pyramid, is the FPN that was proposed by Lin et al. [41] on top of Faster R-CNN [40] network. The general convention, the semantic-rich feature maps are used to enhance spatial-rich feature maps with top-down connected hierarchy. The classical feature pyramid networks have been developed through the years and many different improvements are applied to propose robust and accurate detectors such

as [74, 75, 76] that follow the feature pyramid paradigm.

MatrixNets [74] was proposed as a multi-scale and ratio aware network that is based on FPN [41] architecture. The classical feature pyramid layers are repeated to left and down to create a matrix network with height and width downsampling. The repeated feature maps are derived from diagonal feature maps which have different spatial and semantic information. Also, repeated feature maps have different aspect ratios because their resolutions in specific dimensions are downsampled for row and column of a matrix. The proposed feature-matrix is capable of dealing with extreme aspect ratios with off-diagonal layers that have rectangular shapes. The feature maps with different aspect ratios significantly improve the classical FPN and detection of small objects can be improved by different receptive fields. The novel feature pyramid module of MatrixNets can be used with any type of object detector such as anchor-based or anchor free, one-stage, or two-stage.

HawkNet [75] was proposed to enhance object detection in aerial images based on FPN [41]. The conventional FPN has a top-down pathway between feature maps with lateral connections. The HawkNet removes the lateral connections and up-scale each feature map to equal resolution for concatenating them to construct a top-down feature pyramid system. The up-scaling and concatenation achieve to share more balanced features for each level of feature maps. The semantic-rich feature maps are upsampled to equal size of shallow feature maps and improve their features to increase the performance of small object detection. Moreover, the novel up-sampling method is used rather than the classical interpolation methods such as bilinear or un-pooling and learning-based deconvolution because of their lack of extraction information and corruption on features by checkerboard artifacts respectively. The novel up-scaling module is developed based on sub-pixel convolution and channel unification. The up-scale feature aggregated feature pyramid and efficient up-sample methods significantly improve object detection for aerial imaging which contains many objects that are small in proportion to images.

EfficientDet [76] was proposed as a scalable network with regards to width,

depth, and resolution due to the purpose of usage. It uses EfficientNet which was developed by the same authors as the backbone. The scalability feature facilitates the usage of the network in different domains while preserving performance/accuracy balance. Apart from the scalability of the network, EfficientDet adopts the classical FPN [41] and presents a new FPN module as bidirectional FPN. The bidirectional FPN proposes cross-scale connections similar to NAS-FPN [77] but a much optimized and efficient version to achieve more robust feature fusion and improve the feature layer that has less contribution. Therefore, skip connections are provided between the original feature layer to the fused counterpart to obtain more strong features for final feature maps. The classical FPN fuses feature maps from different resolutions equally but the contribution of features can create imbalance features because the importance of fused features are unknown and degrade the performance of the network. Hence, EfficentDet uses weighted feature fusion methods while features are fused so the network can learn weight parameters and decide which features are more important for detection accuracy. In short, EfficientDet achieves state of art results without losing efficiency in terms of speed and memory with proposed components.

4.2.3.4 Other detectors

MR-CNN[78] was proposed as a region proposal network with integrated multi-scale features. The different layers in depth from VGG-16[26] as backbone are upscaled with deconvolutional layers for concatenation. The fused feature map is processed from the region proposal network to obtain proposals of objects. The final prediction proceeds on features of proposals. The fused feature map is leveraged by semantic and spatial information from different layers. Also, the leveraged feature map enhances multi-scale contextual information and quality of features of object proposals which affect detection performance directly. MR-CNN achieved the state of results on Tsinghua-Tencent 100K [47] dataset.

Liu et al. [79] proposed a two-stage detector which based on Faster R-CNN [40] to improve small vehicle detection. The key idea of research is enhancing

the quality of generated proposals in the RPN network. Thus, Backward Feature Enhancement Network (BFEN) module is designed to advance the feature maps. The BFEN module fused features from layers with different resolutions to generate multi-scale fused feature maps. The shallow layers are supported by deeper layers with semantic information and become more suitable to detect small objects. The multi-scale features maps are used to generate proposals that are passed through a novel Spatial Layout Preserving Network (SLPN) module to refine them for better localization accuracy.

4.2.4 Deformable convolution and pooling

The variable scales and shapes of objects may cause problems that trained object detectors cannot make accurate predictions because generalization capability cannot cover all possible visual appearances. The object detectors are trained to overcome this problem with large data with a variety of data which is supported with augmentation. Deformable convolution and pooling layers are designed to solve this problem from different perspectives that aim to increase the generalization capability of object detectors with adaptive filters and sampling, by Dai et al. [80]. The conventional convolution and pooling operation process fixed grids on images with specified kernel size. Thus, the produced receptive field is limited and static. The static behavior causes to miss semantic information from the image especially for non-rigid or obscure objects. The authors enhanced the conventional convolution and pooling layers that can learn offsets of fixed grid points for adjusting dynamic receptive fields. In other words, a kernel of filters or sampling layers selects the candidate pixels for processing when training continues. The dynamic receptive fields increase the accuracy of localization of objects and produce more descriptive features. In the aspect of small objects, adaptive receptive fields are beneficial to determine them despite their noisy and weak spatial information. The implementation of deformable layers brings negligible overhead because of extra computation that occurs for learnable offsets. Furthermore, deformable layers can be easily replaced with their conventional counterparts in object detectors.

4.2.5 Contextual information

Contextual information, also named contextual reasoning or context priming, helps to increase the detection accuracy of objects in specific environments. In real life, humans perceive objects by their shapes, colors, texture, etc. Also, the human vision system processes the scene and makes connections between objects. The brain of a human learns that some objects inhesion with specific environments or other objects. For example, an airplane usually exists in the sky or airport. The object detection techniques which are especially neural network-based mimics the functionality of the human brain such as a neuron, activations, etc. The researchers have continued to develop this mimicry and have proposed that object detection techniques can be used for contextual information like humans to improve detection. Through the years, three common approaches are developed to describe the contextual relationship of objects: global context, local context, and interactivity in context [10] (Figure 4.2.3). The global context behaves as the external source of the scene and gives information about the scene. It takes to learn context information from looking at the whole image and it classifies the regions of interest which have the potential to comprise objects. The deep learning-based object detectors can achieve global contexting with increasing receptive fields [81], using global pooling operation [82] and recurrent neural networks [83]. The local context aims to use the surrounding area of the object in the scene for achieving detection of it. The first example of using local context was the research of Sinha and Torralba [84] was proposed that local contextual regions of a facial area can increase face detection performance. The big receptive fields and object proposals reveal the local context information for deep learning-based object detectors [10]. The interactivity in context focuses on the interaction of objects in the scene. The related studies [85, 86, 87, 88] have tried to make connections between objects and scenes and determine the constraints and dependencies between them.

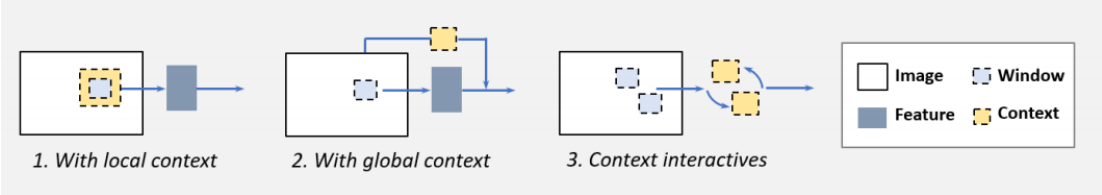


Figure 4.2.3: The different types of contextual priming [10].

In the sight of small object detection, contextual reasoning or information produce a positive effect on detection performance. The small objects can appear in specific environments or other objects because of relationships and dependencies. The detection of small objects with large receptive fields or object proposals is so difficult because of insufficient presence. However, the disadvantage of this situation can be reversed by using contextual relationships. The large objects or specific environments are used to point small objects with developing specialized context-aware modules for deep learning models. In recent years, context-aware deep learning-based object detectors have been developed and achieved decent performance to elevate the detection accuracy of small objects. For instance, authors of this work [89], used the power of contextual information and obtained state of art result for the detection of tiny faces on FDDB [90] and WIDER FACE [53].

Chen et al. [91] proposed an augmented R-CNN to gain the power of contextual information for enhancing small object detection. The context region and proposal region are extracted with parallel networks. The region proposal network is used to extract context regions to encode contextual information from regions that contain small objects. The small objects are extracted with another region proposal network with scalable anchor sizes as a proposal region. The center of context regions is determined from the center of the proposal region of small objects. The extracted regions are processed through convolution layers and extracted same length feature vectors which are concatenated before feeding into the detection network. Augmented R-CNN achieved good results on small object detection when compared with original R-CNN [36] as a baseline.

Inside-Outside Net (ION) [83] was designed to use recurrent neural network blocks for revealing contextual information around objects. The Fast R-CNN is used as a baseline detector and expanded with directional IRNN blocks. The multi-scale features are gathered from the backbone of the baseline detector and concatenated together after normalization is applied. The contextual feature map is obtained through the cascaded IRNN modules and combined with concatenated multi-scale features. The IRNN modules are stacked as pairs and cover all directions of pixels at four dimensionalities which are left, right, up, and down. The produced feature map from IRNNs contains contextual information around objects in those four directions. Each cell on the feature map contains local context information and the output of cells are dependent on other input with help of recurrent blocks.

VSSA-Net [92] was developed to improve the detection of small objects with contextual priming and multi-scale feature enhancement. The study aimed to enhance traffic sign detection as a small object. The VSSA-Net uses MobileNet [34] as a backbone which is a lightweight and fast network with multi-scale input. On the top of MobileNet, a deconvolutional block is deployed with skip connections. The skip connections are used to prevent gradient vanishing and preserve features that are produced by shallow layers. The deconvolutional block amplifies the multi-scale features map and improves semantic-information spatially. Also, it makes the network deeper in a way of obtaining semantic-rich features. The multi-scale and amplified feature maps are fed into the Vertical Spatial Sequence Attention (VSSA) model to extract contextual information. The VSSA model processes feature maps column by column to capture contextual attention vertically and with the use of Long Short-Term Memory (LSTM) module encoding and decoding are applied to these vertical sequences.

[93, 94, 87, 95, 96, 97] are other examples which use context information to improve small object detection.

4.2.6 Super-resolution

Super-resolution refers to generating high-resolution images from low-resolution ones. The generated high-resolution images contain more detailed and strong features. Explained as before, the reason behind the challenge in the detection of small objects is mostly caused by their low-resolution appearance in the scene. The researchers have applied super-resolution techniques to object detection systems and examined that small objects can benefit from super-resolution because of their low-resolution appearance. In recent years, several super-resolution techniques that are based on deep learning have been developed. The Generative Adversarial Network (GAN) become popular for image to image style transferring [98], generating unique images, state learning [99], modification on image [100] and especially for generating super-resolution images [101]. The GAN based super-resolution techniques bring a new dimension to the object detection field and proposed GAN super-resolution and object detection combinations boost the object detection performance exclusively for small objects. The common flow of GAN based super-resolution networks is composed of two networks: discriminator and generator. The generator has to learn to generate super-resolution images from low-resolution counterparts and produce high-resolution images. The discriminator network distinguishes real high-resolution images from high-resolution images which are generated by a generator network. The objective of the GAN algorithm is that the generator learns to produce high-resolution images that are generated for discriminator indiscernible from real ones.

Li et al. [102] proposed GAN based object detector, Perceptual GAN, that aims to increase the detection accuracy of small objects. As a typical GAN network, the proposed network has discriminator and generator parts. The generator which uses a residual neural network is responsible for generating high-resolution features of small objects. The discriminator has two branches: adversarial and perceptual. The adversarial branch discriminates between generated high-resolution small objects features and original large objects features. This discrimination enhances the generation quality of the generator for high-resolution small object features because the generator has to fools the discriminator that the

quality of generated small object features will be very close to the original large ones. The perceptual branch is the classic head part of an object detector and is responsible for object localization and classification. The small objects' features are enhanced to the quality of large objects' features so their weak feature representation and imbalance between large objects are eliminated.

Small Object Detection via Multi-Task Generative Adversarial Network (SOD-MTGAN) was proposed as a specialized object detector for small objects by Bai et al. [103]. The SOD-MTGAN consists of a baseline detector, generator, and discriminator network. The type of baseline detector can be any type such as Faster R-CNN [40] or SSD [5]. The baseline detector is used for generating regions of interest and distinguishing foreground/background objects for the generator and discriminator. The generator takes the low resolution of the original input image and up samples foreground/background small objects to their original resolution. The discriminator discriminates between generated high-resolution objects and original objects. Also, the discriminator predicts the class of objects and the location of objects. As a result of training, the network learns super-resolution and object detection capability. The super-resolved patches boost the small object detection with enhanced spatial information and strengthened feature representation.

There are several other GANs [104, 105, 106] and super-resolution network [107, 108] based object detectors that are specialized for small objects through the use of the power of generating super-resolution images.

4.3 Anchor boxes and loss function

4.3.1 Anchor boxes

The anchor boxes which are also called different names: default boxes, prior boxes, or grid cells, are used in object detection systems to generate predicted bounding boxes and their classes. The design and amount of anchor boxes are important

issues because they have to be chosen to generalize position, size, aspect ratio, and categorization of training data for better regression of predicted bounding boxes and object classes. The classical form of loss function for regression object localization and classification can be formulated as (Figure 4.3.1):

$$L(p, p^*, t, t^*) = L_{cls.}(p, p^*) + \beta I(t) L_{loc.}(t, t^*)$$
$$I(t) = \begin{cases} 1 & \text{IOU}\{a, a^*\} > \eta \\ 0 & \text{else} \end{cases}$$

Figure 4.3.1: The total loss function is a summation of cross-entropy loss for classification and regression loss for object localization [10].

Total loss is a weighted sum of localization and classification loss of predicted objects for ground truth objects. p and p^* are donated for class probabilities of predicted and ground truth objects, t and t^* represent their bounding box information. The localization loss has a condition to related prior boxes as $IOU(a, a^*)$ which means that the anchor box and ground truth box have to match over a threshold that is donated as n. If the IOU of candidate matches does not meet the threshold, the localization of loss has zero multipliers and does not include total loss. In short, unsuitable anchor boxes affect loss function and degrade the learning of object detection models.

The studies on designing anchor boxes proposed several ideas to generate better anchor boxes. The earlier anchor boxes are hand-crafted [5, 109, 40], their sizes and aspect ratios are manually determined. Then, the cluster-based methods [29, 30] are merged, the cluster-based method clusters training data with chosen numbers of anchors according to the input size. The manually chosen and cluster-based predefined anchor boxes are static and require many hyper-parameters. These issues may lead to difficulty in determining parameters and decrease the performance and accuracy of the detector. Recent researches develop anchor-free object detectors and detectors that dynamically learn and refine anchor boxes.

In the aspect of small objects, the manually or cluster-based methods may achieve good results when the large part of the data consists of small objects.

However, the imbalanced data contains a wide range of size, scale, and aspect ratios of objects and static anchor boxes may not cover all objects especially for small objects that are unlikely to match with priors because of their properties. The anchor-free method and dynamic anchor optimization method may lead to boost the performance of small object detection.

4.3.2 Anchor-free detectors

The idea of anchor-free detectors is developed because of several drawbacks of anchor-based detectors. First, predefined or pre-calculated anchor boxes need too many hyper-parameters that have to be fine-tuned or set. The optimized hyper-parameters can make a significant difference in accuracy. For example, different hyper-parameters change the RetinaNet [11] performance in the range of 4% for MS-COCO [33] benchmark in terms of average precision. Thus, taking care of tuning is extra overhead for anchor-based detectors. Secondly, the anchor boxes are chosen to generalize the training dataset and some objects may be affected negatively because of variety in their sizes, scales, or aspect ratios, small objects are one of the examples of these types. Third, to overcome the generalization problem, excessive amounts of anchor boxes are deployed in object detectors but this method leads to an imbalance that focal loss aims to solve, between negative and positive samples. The last drawback is the increasing computational overhead because of matching algorithms that use IOU calculation. The anchor-free detectors try to achieve object detection without anchor boxes such as corner-based [46], center-based [43, 10] or key-points based [110] detectors.Also, hybrid methods are developed as Feature Selective Anchor Free (FSAF) [111] that use the anchor-free module in anchor-based detectors.

FSAF [111] module was designed to achieve anchor-free behavior for one-stage detectors and add dynamic behavior for predefined assignment between anchor boxes and feature maps. The anchor-based detectors that have multiple feature maps in different scales are designed to learn each level of the feature map and learn specific objects that are related to the size of the feature map. To achieve

this, the size of anchor boxes for each level are constrained by hand-crafted rules. The limitation of setting this constraint is that some objects may not be suitable for detection in feature level which is assigned as a prior. FSAF solves this limitation with a feature selection method that allows objects to learn the best feature level in the training phase. FSAF uses regression to produce 4 offset maps to obtain bounding box locations in an anchor-free way. FSAF is applied as a module to any one-stage detectors that have a feature pyramid or detection pyramid. Also, FSAF can work independently or with anchor-based modules jointly.

FCOS [112] was proposed as an anchor-free object detector which is similar to RetiaNet [11] and detection operates analogously to a segmentation. The per-pixel classification is adapted to predicting bounding boxes of objects. Each pixel location on feature maps is projected to ground-truths in the input image and ground-truth boxes are used like anchor-boxes. If the projected location is near to the center of the receptive area and this point is included in the ground truth box, the point is taken as a reference point to make a regression to corners of ground truth boxes. The regression process tries to estimate four distances from the reference point and calculate the minimum bounding box that covers objects. The anchor-free object localization works on multi-level prediction with FPN [41] with some constraints to avoid compute regression of negative samples.

4.3.3 Anchor optimization

Apart from anchor-free methods, several optimization methods are developed for anchor-based object detectors. The methods aim to generate anchor boxes better than manually driven and naive clustering methods. The studies on anchor optimization have several types such as novel optimization methods [113], anchor refining modules [114] or self-learning anchors modules [115]. The optimization methods facilitate the training phase in the aspect of hyper-parameters can be chosen more dynamically and efficiently than traditional greedy methods such as grid search. The anchor refining modules work together with predefined anchors

and generate dynamic anchor functions to obtain custom anchors for better detection. The self-learning anchor modules learn anchor shapes during training and optimize the anchor sizes dynamically. The advantage of anchor optimization modules is that most of them integrate into state of art detectors easily because of plug-in structure and their computation cost can be negligible besides improvement in accuracy.

4.3.4 Loss function

The loss function is a very important concept in deep learning-based applications. The deep learning models do forward and backward calculations on their layers to calculate estimated outputs and optimal parameters for the layers. The forward calculation means that elements of the network process the input and produce actual output in one pass. The actual output is desired to be close to the ground truth value of the training set. The criterion of closeness is measured by the loss function which calculates the error between desired and predicted value. In other terms, the loss function is the function that is desired to minimize through the learning process because when the ground-truth value matches with the predicted one, the closeness is maximized and error becomes zero. In the learning process, the loss function guides the back-propagation(backward calculations) from starting calculation of error's derivative and each layer of network calculates derivative itself in a manner of differential functions. The derivatives of layers show how much changes have to be done for the parameters of layers. For each forward and backward pass, the loss function shows the performance of current parameters and gives information to the back-propagation phase. Also, the loss function is mentioned as an objective function that has to be maximized in contrast to the loss function. Furthermore, the selection of loss function has a place in network performance because an unsuitable loss function produces poor results. There are several loss functions used in deep learning for both regression and classification algorithms. Mean squared error, mean absolute error, root mean squared error, and quantile loss are some examples of regression problems. Log loss(cross-entropy loss), hinge loss, exponential loss, and focal loss [11] are considered as

examples of classification problems. In the perspective of a small object challenge, the focal loss achieves good performance to increase the detection accuracy.

4.3.5 Focal loss

The foreground/background class imbalance is one of the challenges in object detection that has to be solved. The object detectors suffer from this imbalance during the training phase. The two-stage detectors are more robust to this problem because they generate region proposals and these proposals are used in classifiers sparsely. The region proposals are selected candidate regions of interests and filtered out negative candidates and pass through the candidates into the second stage which classifies regions as foreground/background objects. The region proposal generators such as Selective Search [37], DeepMask [116], EdgeBoxes [117] and RPN [40] eliminate the most of negative samples. Moreover, online hard example mining [44] and some generalized functions like fixed foreground/background ratios decrease the imbalance between foreground/background classes.

However, one-stage detectors are designed to classify objects on the image directly with a dense sampling method. The dense sampling method leads to generating excessive negative samples which are regions that do not include any objects. The imbalance between negative and positive samples causes several problems in training especially and affects the accuracy of the final model. The quality of training may have fallen because the high contribution of negative samples could not boost the learning rate of object detectors. In contrast, useless information slows the training and decreases the accuracy like a noise. Hence, some one-stage detectors use hard negative mining [44] and bootstrapping [118] to obviate the imbalance but these methods bring inefficiency in terms of computation while their weak contribution to solve the imbalance.

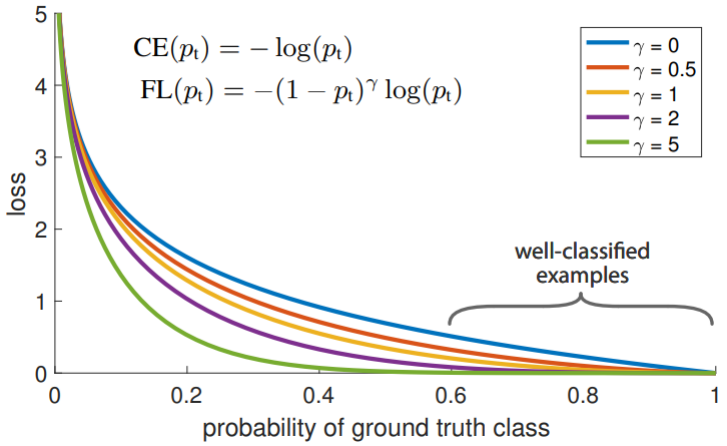


Figure 4.3.2: The relationship between cross entropy and focal loss with respect to γ. The loss of well-classified examples can dominate the total loss when classic cross entropy loss is used. The γ factor reduces the contribution of easy negatives and gives more weight to hard negatives to balance the training [11].

RetinaNet [11] was proposed as a one-stage detector with a novel loss function that was named Focal Loss. The focal loss was designed to solve foreground/background class imbalance with a dynamically weighted loss function which reduces the contagion of negative samples. The focal loss is derived from cross-entropy loss for classification. The mathematical formulation of cross-entropy loss for binary classification is expressed in Figure 4.3.3

$$\mathrm{CE}(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise.} \end{cases}$$

Figure 4.3.3: The mathematical formulation of cross entropy loss for binary classification [11].

y is donated for ground-truth class and p is for estimation probability of when $y = 1$. When simply define p_t as 4.3.4 and replace with (p, y) we obtain a new version of cross entropy loss as Figure 4.3.5:

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise,} \end{cases}$$

Figure 4.3.4: The expression of p_t [11].

$$\mathrm{CE}(p, y) = \mathrm{CE}(p_t) = -\log(p_t).$$

Figure 4.3.5: The cross entropy loss when we define p_t as Figure 4.3.4 [11].

As shown as Figure 4.3.2 easy classified examples ($p_t >> .5$) may overwhelm the less frequent classes with their large quantities which cause a huge accumulation of small errors. To solve this imbalance, α weight factor can be assigned to the formula and it can be related to the class frequency or can be a hyper-parameter that has to be determined before actual training (Figure 4.3.6).

$$\mathrm{CE}(p_t) = -\alpha_t \log(p_t).$$

Figure 4.3.6: The α-balanced cross entropy loss [11].

However, α only deals with a balance of positive/negative examples and does not care about easy/hard examples. Easy classified examples can easily ruin the cross-entropy loss because they dominate the loss and gradients with their quantities and they are treated as the same as hard examples. The focal loss is designed to focus on training hard negatives so the cross-entropy is modified with $(1 - p_t)\gamma$ and γ is the tunable parameter and bigger than zero. The focal loss is represented as 4.3.7:

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

Figure 4.3.7: The focal loss[11].

When we look at the formulation of focal loss, we can observe that when an object is misclassified and has low p_t, the loss does not change so much but when

p_t close to 1 and a well-classified object has a lower weight and easy examples can be differentiated in the loss function. γ is the rate of down-weighted and the authors state that $\gamma = 2$ is the best value in their experiments.

In conclusion, the performance of small object detection suffers class imbalance because they are hard to differentiate from the background and their numbers can be lower than medium and large objects. Thus, the focal loss can improve the accuracy of small object detection in a way to penalize the hard negatives which include the small objects because small objects are hard to detect and their misclassification has to affect loss other than easy negatives.

Chapter 5

Results and Discussions

In this research, the evaluation and comparison of approaches into four main groups which we introduce for fairness: intra-comparison, intercomparison, global comparison, and a group of exclusion from the comparison.

The intra-comparisons are done for pre-post processing, the effect of backbone networks, and regularizing hyperparameters or priors on object detection networks. In this type of comparison base network which produces results are considered as base scores for benchmarking, and regulated base networks have almost identical network architecture. Also, intra-comparisons are presented within sections that explain the methodology of approaches.

The intercomparison is used for network modifications such as DSSD [54] and FSSD [55] which are derived from SSD [5] as baseline detector. The intercomparison serves to show the effect of modifications on the baseline detector and make the trade-off analysis.

In global comparison, the approaches which share the same or similar version of training and test dataset, are compared with respect to their average precision on small objects with their inference speed.

Finally, some approaches are dedicated to very specific parts of small object

detection such as tiny face detection, traffic sign detection, etc or they do not share common datasets that are used for comparison in this study. The contributions of these approaches are valuable for the detection of small object challenges but we cannot directly compare them with other works. Hence, this type of approach is not evaluated and compared with other works.

5.1 Comparison on MS-COCO

The MS-COCO [33] dataset is used for comparisons. The evaluation results of MS-COCO are obtained from other researches or surveys. To obtain fair comparison the versions of training and test dataset are specified for MS-COCO. Furthermore, the inference speed of detectors are given as frame per second and the specifications of GPUs are provided for accurate comparison (Table 5.1). The multiplier property is given in Table 5.1 to compare inference speed of detectors that are evaluated in different GPUs and the multiplier ratios are approximate. The real-time requirement is specified as 30 FPS with a batch size of 1. The general evaluation criteria of MS-COCO is given in Figure 5.1.1 and we only indicate the $AP@0.5:0.95(S)$ in the comparison table (5.2, 5.3, 5.4) because the main focus on improvement in small object detection but the other metrics are added to Table A.1, A.2 and A.3 in Appendix A for supplementary information. The AP_s is used instead of $AP@0.5:0.95(S)$ for simplicity in our comparison sections.

Table 5.1: Comparative GPU Specifications

GPU Model	CUDA Cores	FP32(float)	Multiplier
(K)Nvidia K40	2880	5.046 TFLOPS	X
(M)Nvidia Titan X	3072	6.691 TFLOPS	3.5X
(M)Nvidia M40	3072	6.844 TFLOPS	3.5X
(P)Nvidia 1080 Ti	3584	11.34 TFLOPS	6X
(P)Nvidia Titan X	3584	10.97 TFLOPS	6X
(V)Nvidia V100	5120	14.13 TFLOPS	10X

MSCOCO	• AP_{coco}: mAP averaged over ten Ω: $\{0.5 : 0.05 : 0.95\}$; • AP_{50}: mAP at 0.50 IoU threshold; • AP_{75}: mAP at 0.75 IoU threshold; • AP_S: AP_{coco} for small objects of area smaller than 32^2; • AP_M: AP_{coco} for objects of area between 32^2 and 96^2; • AP_L: AP_{coco} for large objects of area bigger than 96^2;

Figure 5.1.1: The MS-COCO explanation of evaluation metrics [9].

5.1.1 Comparison table

Each method in Table 5.2, 5.3 and 5.4 has a citation reference number of corresponding original paper and evaluation results are taken from the corresponding paper except for methods which have two citation reference number. The result of these methods is taken from the second citation reference number and the first number indicates the original paper. Moreover, some methods have some indicators that are (D) and (MS). The (D) indicates that the proposed method uses deformable convolution layers. The (MS) indicates that the proposed method uses an image pyramid paradigm with the default one.

Table 5.2: MS-COCO Evaluation Results

Method	Input Size	Backbone	AP_S	Train Set	Test Set	GPU	FPS
Faster R-CNN [40][55]	600x1000	VGG-16	7,7	trainval	test-dev	(M)Nvidia Titan X	7
Faster R-CNN [40][109]	600x1000	ResNet101	6,6	train	val	(K)Nvidia K40	2
Faster R-CNN-FPN [41][119]	600x1000	ResNet50	21,3	trainval35k	test-dev	(M)Nvidia M40	7
Faster R-CNN-FPN [41][69]	600x1000	ResNet101	18,2	trainval35k	test-dev	(M)Nvidia M40	6
TridentNet [64]	-	ResNet-101	23,9	trainval35k	test-dev	(P)Nvidia Titan X	2,7
TridentNet (D) [64]	-	ResNet-101	28	trainval35k	test-dev	(P)Nvidia Titan X	1,3
TridentNet (MS)(D) [64]	-	ResNet-101	31,8	trainval35k	test-dev	-	-
SNIPER (D) [60]	-	Resnet101	30,9	trainval35k	test-dev	(V)Nvidia V100	5
ION [83]	-	VGG-16	14,5	trainval35k	test-dev	(M)Nvidia Titan X	1
YOLOv2 [29][68]	544x544	Darknet-19	5	trainval35k	test-dev	(M)Nvidia Titan X	40
YOLOv3 [30][1]	320x320	Darknet-53	11,9	trainval35k	test-dev	(M)Nvidia M40	45
YOLOv3 [30][1]	416x416	Darknet-53	15,2	trainval35k	test-dev	(M)Nvidia M40	35
YOLOv3 [30][1]	608x608	Darknet-53	18,3	trainval35k	test-dev	(M)Nvidia M40	20
YOLOv3 [30][1]	608x608	Darknet-53	20,6	trainval35k	test-dev	(M)Nvidia M40	20
YOLOv4 [1]	416x416	CSPDarknet-53	20,4	trainval35k	test-dev	(M)Nvidia M40	38
YOLOv4 [1]	512x512	CSPDarknet-53	24,3	trainval35k	test-dev	(M)Nvidia M40	31
YOLOv4 [1]	608x608	CSPDarknet-53	26,7	trainval35k	test-dev	(M)Nvidia M40	23
SSD [5][1]	300x300	VGG-16	6,6	trainval35k	test-dev	(M)Nvidia M40	43
SDD [5][1]	512x512	VGG-16	10,9	trainval35k	test-dev	(M)Nvidia M40	22
SSD [5][54]	321x321	Resnet101	6,2	trainval35k	test-dev	(M)Nvidia Titan X	11,2
SDD [5][54]	513x513	Resnet101	10,2	trainval35k	test-dev	(M)Nvidia Titan X	6,8

Table 5.3: Continuation of Table 5.2

Method	Input Size	Backbone	AP_S	Train Set	Test Set	GPU	FPS
DSSD [54]	321x321	Resnet101	7,4	trainval35k	test-dev	(M)Nvidia Titan X	9,5
DSSD [54]	513x513	Resnet101	13	trainval35k	test-dev	(M)Nvidia Titan X	5,5
FSSD [55]	300x300	VGG-16	8,7	trainval35k	test-dev	(P)Nvidia 1080 Ti	65,8
FSSD [55]	512x512	VGG-16	14,2	trainval35k	test-dev	(P)Nvidia 1080 Ti	35,7
CFE-SSDv1 [68]	300x300	VGG-16	10,6	trainval35k	test-dev	(M)Nvidia Titan X	42,2
CFE-SSDv1 [68]	512x512	VGG-16	16,2	trainval35k	test-dev	(M)Nvidia Titan X	22
CFE-SSDv2 [68]	300x300	VGG-16	12,1	trainval35k	test-dev	(M)Nvidia Titan X	41,5
CFE-SSDv2 [68]	512x512	VGG-16	17,5	trainval35k	test-dev	(M)Nvidia Titan X	21,2
CFE-SSDv2 [68]	512x512	Resnet101	21,2	trainval35k	test-dev	(M)Nvidia Titan X	11
M2Det [69]	320x320	VGG-16	14,4	trainval35k	test-dev	(P)Nvidia Titan X	33,4
M2Det [69]	512x512	VGG-16	18,4	trainval35k	test-dev	(P)Nvidia Titan X	18
M2Det [69]	320x320	Resnet101	14,8	trainval35k	test-dev	(P)Nvidia Titan X	21,7
M2Det [69]	512x512	Resnet101	20,5	trainval35k	test-dev	(P)Nvidia Titan X	15,8
M2Det [69]	800x800	VGG-16	22,1	trainval35k	test-dev	(P)Nvidia Titan X	11,8
EFGRNet [65]	320x320	VGG-16	13,4	trainval35k	test-dev	(P)Nvidia Titan X	47,6
EFGRNet [65]	512x512	VGG-16	19,7	trainval35k	test-dev	(P)Nvidia Titan X	25,7
EFGRNet [65]	512x512	Resnet101	17,8	trainval35k	test-dev	(P)Nvidia Titan X	21,7
Pang et al. SSD (MS) [62]	300x300	VGG-16	10,9	trainval	test-dev	(P)Nvidia Titan X	71
Pang et al. SSD (MS) [62]	512x512	VGG-16	18,3	trainval	test-dev	(P)Nvidia Titan X	34
IPG RCNN [63]	-	IPG-Net101	26,6	trainval35k	test-dev	-	-
CornerNet511 [46]	512x512	Hourglass-104	19,1	trainval35k	test-dev	(M)Nvidia Titan X	4,4

Table 5.4: Continuation of Table 5.3

Method	Input Size	Backbone	AP_S	Train Set	Test Set	GPU	FPS
MatrixNets [74]	900x900	ResNet-152	25,9	trainval35k	test-dev	(P)Nvidia Titan X	2,8
MatrixNets (MS) [74]	-	ResNet-152	29,7	trainval35k	test-dev	-	-
FSAF [111]	800x800	ResNet-101	24	trainval35k	test-dev	(P)Nvidia Titan X	5,5
FSAF (MS) [111]	-	ResNet-101	27,8	trainval35k	test-dev	-	-
FSAF [111]	800x800	ResNeXt-101	26,6	trainval35k	test-dev	(P)Nvidia Titan X	2,76
FSAF (MS) [111]	-	ResNeXt-101	29,7	trainval35k	test-dev	-	-
FCOS* [112]	-	ResNet-101	31	trainval35k	test-dev	(P)Nvidia 1080 Ti	38
FCOS [112]	-	ResNeXt-32x8d-101	32	trainval35k	test-dev	(P)Nvidia 1080 Ti	-
FCOS (D) [112]	-	ResNeXt-32x8d-101	33,2	trainval35k	test-dev	(P)Nvidia 1080 Ti	-
RetinaNet [11]	500x500	ResNet-50	13,9	trainval35k	test-dev	(M)Nvidia Titan X	13,9
RetinaNet [11]	500x500	ResNet-101	14,7	trainval35k	test-dev	(M)Nvidia Titan X	11,1
RetinaNet [11]	800x800	ResNet-50	18,9	trainval35k	test-dev	(M)Nvidia Titan X	6,5
RetinaNet [11]	800x800	ResNet-101	20,2	trainval35k	test-dev	(M)Nvidia Titan X	5,1
EfficientDet-D0 [76][1]	512x512	Efficient-B0	12	trainval35k	test-dev	(V)Nvidia V100	62,5
EfficientDet-D1 [76][1]	640x640	Efficient-B1	17,9	trainval35k	test-dev	(V)Nvidia V100	50
EfficientDet-D2 [76][1]	768x768	Efficient-B2	22,5	trainval35k	test-dev	(V)Nvidia V100	41,7
EfficientDet-D3 [76][1]	896x896	Efficient-B3	26,6	trainval35k	test-dev	(V)Nvidia V100	23,8
SOD-MTGAN [103]	-	ResNet-101	24,7	trainval35k	test-dev	-	-

5.1.2 Comparison of backbones(intra-comparison)

The effects of backbones are changed by the architectural design of object detectors and themselves. The features of small objects are preserved in shallow layers mostly. The backbone detectors can be strengthened with group convolution, deformable convolution, and pooling layer and skip connections. If the detector produces a single feature map without feature fusion or has a prediction pyramid system, the detector may produce less accurate results with deeper simpler backbones for small objects. For example, Faster R-CNN [40] with VGG-16 [26] performs better than ResNet-101 [28] which is deeper than VGG-16. The same behavior repeats between VGG-16 and Resnet-101 for SSD [5]. The features of small objects vanish in deeper feature maps and shallow feature maps are produced in deeper layers without any feature aggregation from shallow layers. On the other hand, the feature aggregation, more advanced backbones, and improved loss functions increase the effectiveness of deeper networks and leverage the accuracy in the detection of small objects. The shallow layers can be supported with semantic information or deeper layers are powered with spatial-rich information. For example, CFE-SSDv2 [68] performs better with Resnet-101 than VGG-16 because feature aggregation modules are used in architecture. The same behavior can be observed between ResNet-50 and Resnet-101 for RetinaNet [11] or VGG-16 and Resnet-101 for M2Det [69]. In short, we could not specify a certain rule to choose the backbone to obtain better accuracy, we may consider interactions of components in the detector.

5.1.3 Inter-comparison

5.1.3.1 Yolo variants

If YOLOv2 [29] is considered a base network, YOLOv3 [30] outperforms YOLOv2 with a prediction pyramid paradigm and stronger backbone. The lowest resolution configuration of YOLOv3 doubles the accuracy of small object detection without sacrificing inference speed. YOLOv4 [1] has a SPP module that adds an

extra AP_S to YOLOv3 without an increase in inference speed. The YOLOv4 increases AP_S about 30% with slightly better inference time. The better backbone and neck modules highlight the YOLOv4 with real-time inference speed in the YOLO family.

5.1.3.2 SSD variants

There are many object detectors developed on top of SSD [5]. DSSD [54] uses deconvolutional blocks to increase accuracy in the detection of small objects but if we consider SSD-512 with VGG-16 [26] as a baseline, DSSD only puts an extra 2,1% AP_S with a quarter of inference speed. The complex deconvolutional system cannot achieve a balanced trade-off between accuracy and speed. If FSSD [55] is evaluated with the same baseline, FSSD achieves a much better accuracy/speed trade-off than DSSD and outperforms the SSD with extra 3,3% precision and a slight decrease in inference speed. CFE-SSDv1 and CFE-SSDv2 [68] with VGG-16 backbone increase AP_S 5,3% and 6,6% respectively with achieving the same inference speed. CFE-SSDv2 with ResNet-101 [28] backbone outperforms the same baseline with nearly doubling AP_S but inference speed halves. If Nvidia Titan X (Pascal) gives nearly 1.7x more frame per second than Nvidia M40(Maxwell), M2Det-512 [69] with Resnet-101 backbone nearly doubles the AP_S but fall behind CFE-SSDv2 with ResNet-101 backbone in terms of precision as 0,7% and inference speed as 18%. EFGRNet-512 [65] with VGG-16 backbone could not outperform CFE-SSDv2 with VGG-16 backbone because of a slight improvement in the precision of small objects but a huge difference in inference time negatively. The winner of SSD variants is an object detector proposed by Pang et al. [62] with the best trade-off between accuracy and inference time. The proposed network by Pang et al. with VGG-16 backbone and 512x512 input resolution, achieves 18,3% AP_S with a slight decrease in inference when comparing to baseline. The best average precision in small objects is achieved by M2Det-800 with VGG-16 backbone as 22,1% but inference speed decreases drastically.

5.1.3.3 FPN variants

The first feature pyramid network was the modification of Faster R-CNN [40] with a feature pyramid module. As a baseline, Faster R-CNN with VGG-16 [26] and ResNet-101 [28] backbone achieve 6,6% and 7,7% AP_S and their FPS measured as 7 and 2 respectively. After improvement with FPN, Faster R-CNN via ResNet-101 backbone triples the mean average precision for small objects with a slight decrease in inference speed. Moreover, Faster R-CNN via ResNet-50 and FPN achieve more than three times than baseline without losing any inference speed. The CornetNet [46] is another object detector built with the FPN module and uses the Hourglass-104 backbone. The CornerNet ,anchor-free one-stage detector, has 19,1% AP_S with 4,4 FPS inference speed. When we compare with Faster R-CNN-FPN-ResNet-50 [41], we expect much more inference speed because of the one-stage property but it falls behind in comparison. The RetinaNet [11], owner of focal loss, uses FPN with ResNet-50 and ResNet-101 and achieves 18,9% and 20,2% AP_S , 6,5 and 5,1 FPS as inference speed. The results of RetinaNet are better than CornerNet and compete head to head with Faster R-CNN-FPN. The other one-stage detector, MatrixNets [74], uses FPN with ResNet-152 and very high input resolution as 900x900 pixels. The results of MatrixNets for AP_S are 25,9% (without image pyramid) and 29,7% (with image pyramid). The results outperform Faster R-CNN-FPN as baseline and RetinaNet but inference speed decreases drastically. The EfficientDet [76] and FCOS [112] are one-stage detectors that use different versions of FPN as bidirectional FPN. The FCOS is an anchor-free detector and achieves 31% AP_S with ResNet-101 backbone and has 38 FPS which were reported when using ResNet-50. The ResNeXt-32x8d-101 [31] version of FCOS also has 32% AP_S and the same version that uses deformable convolution has 33,2%, the highest average precision for small objects in our comparison. The expected inference speed for ResNet-101 and ResNeXt-32x8d-101 are 17 , 10 FPS respectively. The inference speed does not meet the real-time requirements but outperforms the FPN category with high average precision. On the other hand, EfficientDet is a scalable network and the trade-off between inference speed and accuracy can be set for requirements. The accuracy of EfficentDet is a bit behind the FCOS in contrast to inference speed.

5.1.4 The global comparison

The YOLOv4 [1], Pang et al. [62], EfficientDet [76], and FCOS [112] outperform their counterparts in terms of small object detection accuracy and inference performance with respect to the output of our inter-comparisons for the MS-COCO dataset. These detectors are named as leading detectors for simplicity in this section. Several object detectors are not included in inter-comparisons, compared in this section. The ION uses contextual priming with the fashion of Faster R-CNN [40] produces 14,5% AP_S and 1 FPS. This result is far away from Faster R-CNN-FPN [41] and the leading detectors. The TridentNet [64] which has a different prediction pyramid system with branching achieves good results for detecting small objects and 23,9% AP_S without using deformable convolutional layers and 28% with using deformable convolutional layers. Also, TridentNet can achieve 31,8% AP_S by using the image pyramid extra. However, all three results are higher than average, even close to top results, the inference speed of the detector is so slow, about 1-3 frames per second. The similar results repeated with FSAF [111], hybrid anchor-free object detector, achieve accuracy above the average but suffer in terms of inference speed. The image pyramid system proves the effectiveness of detecting small objects with higher average precision scores like SNIPER [60] which has 30,9% AP_S but the inference speed falls behind the real-time requirements. In conclusion, FCOS, SNIPER, TridentNet, FSAF, and MatrixNets [74] are prominent detectors with regards to AP_S but they require too much power of computation. The YOLOv4, Pang et al. [62] and EfficientDet has a very satisfactory balance of accuracy/performance trade-off and they can be used in real world scenarios. Deformable convolutional layers, feature aggregation modules, and feature pyramid modules can play a very significant role in small object detection challenges without sacrificing computational power.

5.2 Comparison on VisDrone-DET

The VisDrone-DET [8] dataset is used for comparison to observe results of custom dataset that consist mostly of small objects and has limited training data. The evaluation results that are not indicated with '$*$' are obtained from our simulations. Source codes and trained models can be seen in this repository [1]. The other results that are indicated with '$*$' are obtained by other studies. The difference between other studies and our experiments is the dataset that used for testing. We use validation datasets and other studies use testing dataset for testing phase. Furthermore, the all object detectors that are covered in our study could not included in this comparison because of our limited time and resources for training every object detector. Also, the VisDrone-DET dataset can be considered as a recent dataset and the amount of usage in studies are low. The results are simulations can be incomplete in terms of training procedure and experiments with different hyper-parameters may generate more accurate results. The goal of these experiments is showing the performance of object detectors when trained with custom dataset that small objects have a majority in distribution of dataset and has limited training data. The general evaluation criteria of VisDrone-DET is similar to MS-COCO [33] but it does not use average precision metrics for categorized objects by sizes. Hence, we only indicate the AP@0.5 : 0.95, AP@0.5, AP@0.75 in the comparison table (5.6, 5.7) because we mentioned the reason before in Section 3.1 that almost all objects are considered as small objects in VisDrone-DET dataset. The AP is used instead of AP@0.5 : 0.95 for simplicity in our comparison sections. We use GPUs from Google Colaboratory and local system that has Nvidia RTX 2080 Ti as aGPU and Intel i9-9900k as a Central Processing Unit (CPU) for our experiments. The inference speed of object detectors that are trained by Nvidia RTX 2080 Ti, are stated in comparison tables. On the other side, the inference speed of our experiments that are trained by Google Colaboratory, are not stated. The comparison tables of MS-COCO (5.2, 5.3, 5.4) can be referred for inference speed of experiments are not stated in Table 5.6. The comparison of MS-COCO are divided into three category as intra,inter and global comparison but we do only global comparison for VisDrone-DET because

[1] https://github.com/melikdaye/MSThesis_small_objects_visdrone

the number of results are not sufficient and large as in Tables (5.2, 5.3, 5.4).

5.2.1 Comparison table

Each method in Table 5.6 and 5.7 has a citation reference number of corresponding original paper. The evaluation results of method that indicated with '$*$', are taken from the corresponding paper except for methods which have two citation reference number. The result of these methods is taken from the second citation reference number and the first number indicates the original paper. Other methods that are not indicated with '$*$', have only citation reference number of corresponding original paper and the evaluation results are obtained from our simulations. Object detectors that have inference speed values and trained by us, are tested with Nvidia RTX 2080 Ti and Intel i9-9900k. Object detectors that indicated with '$*$' and have inference speed values, have specification Table 5.5 to indicate properties of computer that used for training and testing. The specifications as shown int Table 5.5 are obtained from [120, 121].

Table 5.5: System Specifications of Cited Results

Method	GPU	CPU	FPS
TridentNet* [64][121]	RTX 2080Ti	Intel E5-2620v4@2.10GHz	0.2
CFE-SSDv2* [68][120]	GTX Titan XP	-	1
Faster R-CNN* [40][120]	GTX Titan X	-	7

Table 5.6: VisDrone-DET Evaluation Results

Method	Input Size	Backbone	*AP*	*AP*@0.5	*AP*@0.75	FPS
YOLOv3 [30]	320x320	Darknet53	0,118	0,246	0,104	78
YOLOv3 [30]	416x416	Darknet53	0,158	0,320	0,14	66
YOLOv3 [30]	608x608	Darknet53	0,212	0,420	0,19	44
YOLOv3 [30]	608x608	Resnet-50	0,198	0,395	0,18	44
YOLOv3 [30]	608x608	Resnet-101	0,111	0,245	0,093	35
YOLOv3 [30]	608x608	Resnet-152	0,105	0,211	0,97	28
YOLOv4 [1]	320x320	CSPDarknet53	0,160	0,305	0,15	54
YOLOv4 [1]	416x416	CSPDarknet53	0,207	0,387	0,195	44
YOLOv4 [1]	608x608	CSPDarknet53	0,267	0,483	0,261	31
SSD [5]	300x300	VGG-16	0,045	0,092	0,042	38
SSD [5]	512x512	VGG-16	0,076	0,141	0,074	16
DSSD [54]	321x321	Resnet-101	0,076	0,154	0,067	-
FSSD [55]	300x300	VGG-16	0,052	0,104	0,048	-
FSSD [55]	512x512	VGG-16	0,128	0,239	0,121	-
Feature-Fused-SSD [67]	300x300	VGG-16	0,06	0,118	0,054	-
Feature-Fused-SSD [67]	512x512	VGG-16	0,088	0,17	0,087	-
EFGR-Net [65]	512x512	VGG-16	0,189	0,341	0,187	-
FSAF [111]	800x800	Resnet-101	0,16	0,293	0,158	-
TridentNet [64]	-	Resnet 101	0,13	0,262	0,114	-
MatrixNets [74]	900x900	Resnet50	0,162	0,274	0,163	-
YOLO-Tiny [29]	416x416	Darknet19	0,04	0,102	0,026	125

Table 5.7: Continuation of Table 5.6

Method	Input Size	Backbone	AP	AP@0.5	AP@0.75	FPS
CornerNet* [46][121]	512x512	Hourglass-104	0,174	0,341	0,157	-
Faster R-CNN-FPN* [41][121]	600x1000	ResNet101	0,165	0,322	0,149	-
RetinaNet* [11][121]	500x500	ResNet101	0,118	0,231	0,116	-
TridentNet* [64][121]	-	Resnet 101	0,225	0,432	0,205	0.2
CFE-SSDv2* [68][120]	2200x2200	VGG-16	0,264	0,473	0,265	1
Faster R-CNN* [40][120]	600x1000	VGG-16	0,035	0,087	0,024	7

5.2.2 The global comparison

We have three different metrics in comparison table (5.6, 5.7) : AP, AP@0.5, AP@0.75. The AP is taken as reference metric for comparing methods. If we start from bottom and go trough the method in terms of accuracy and inference speed. Faster R-CNN [40] has worst accuracy. Also, when accuracy is low, we except better trade-off between accuracy and inference speed but the Faster R-CNN has only 7 FPS, hence it also produce bad trade-off ratio. When we look at second worst method in terms of accuracy, YOLO-Tiny has slightly better accuracy than Faster R-CNN but it has very high inference speed so it produces reasonable accuracy in terms of trade-off between accuracy and inference speed. SSD [5] variants produce unsatisfactory results except CFE-SSDv2 [68] and EFGR-Net [65]. CFE-SSDv2 has high accuracy but it uses very high resolution image so inference speed is very low. The result of CFE-SSDv2 does not show the real power of method because it is boosted by size of input and inference speed is not practical. Faster R-CNN-FPN [41], EFGR-Net, FSAF [111], RetinaNet [11], MatrixNets [74], CornerNet [46] produce moderate results because they produce average accuracy but the drastic decrease in inference speed makes the methods inefficient. TridentNet [64] has two results in the table. The higher result in terms of accuracy is obtained from external paper. We show both results to criticise an our insufficient training. Thus, the higher result is taken into consideration for comparison. Result of TridentNet is similar to CFE-SSDv2, hence it is also impractical method. Variants of YOLOv3 [30] and YOLOv4 [1] have reasonable trade-off between accuracy and inference speed. The variants of YOLOv3 that use ResNet family produce slightly worse results than their counterparts which use Darknet53 as a backbone. When we compare YOLOv3 and YOLOv4, YOLOv4 has a top accuracy and real time inference when input size is 608x608 but YOLOv3 with Darknet-53 can be also chosen with different input settings to get higher inference speed and above the average accuracy within the scope of the comparison table.

Chapter 6

Conclusion and Future Work

In this thesis, small object detection is examined comprehensively in terms of drawbacks, datasets, and methods. The methods are categorized into general approaches that are addressed to drawbacks. We aggregate proposed approaches from recent studies and review them in detail. We try to cover each part of object detection that leads to solving this challenge. The approaches which increase the accuracy of small object detection for deep learning methods focus on three stages of object detector. First, we examine pre/post-processing techniques that are not directly related to the architecture of the detector. The augmentation and pre-trained models improve the training stage and image tiling increases the spatial resolution of small objects by making them not affected by image resizing. Second, the architectural modification or design for object detectors elevates the accuracy of the detector significantly. The design of the backbone, neck, and prediction module changes the feature representation of small objects and facilitates the learning process of the network. The deeper backbones produce better results in the general convention but for small objects, this can be the opposite because of the enlarged receptive field. The more complex backbones may resolve this problem with skip connections, group convolution, and deformable convolution/pooling layers. The neck parts or feature enhancement modules gather different feature types to fuse them for more descriptive features.

Moreover, contextual reasoning and super-resolution techniques also are considered architectural approaches. The contextual reasoning exploits the relationship between features of small objects and their environments to increase the accuracy of their detection. The super-resolution techniques use up-sampling method such as general adversarial networks to increase spatial information of small objects to improve detection. In last, optimizing hyper-parameters and loss function may huge difference in detection accuracy for small objects. The focal loss aims to solve class imbalance which is one of the reasons for drawbacks for small objects with tuning training loss on behalf of objects that hard to learn by the network. The anchor boxes are an important concept for object detectors and their optimization before training improves the accuracy as well. Furthermore, anchor boxes can be learned dynamically in the training phase or they are not used in total such as anchor-free detectors.

In our comparison, we try to prove the effectiveness of all method that mentioned in the previous paragraph and we discuss their effectiveness because each method has some advantages and disadvantages. The general look of view advantage of each method is simply increasing in small object detection accuracy. On the other side, disadvantages are more about computation cost in terms of inference speed and memory. In our thesis, we only focus on inference speed. Therefore, we compare methods for their accuracy/performance trade-off. The methods that increase accuracy slightly but decrease performance significantly may be categorized as unsuitable for real-world scenarios. The methods which are more suitable for real-world scenarios have well-balanced accuracy/performance trade-off with above the average accuracy and real-time inference speed. Also, some methods improve accuracy significantly but they operate in low inference speed. These methods can be used for applications that require high accuracy with high-performance computing devices.

For future work, this survey can be extended with experiments that combine explained approaches differently to show the effectiveness of combinations on a custom dataset. Each combination will be evaluated and compared in terms of accuracy, inference speed, memory consumption, and utilizing a computing device.

Bibliography

[1] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[2] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, p. 128837–128868, 2019.

[3] G. Chen, H. Wang, K. Chen, Z. Li, Z. Song, Y. Liu, W. Chen, and A. Knoll, "A survey of the four pillars for small object detection: Multiscale representation, contextual information, super-resolution, and region proposal," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.

[4] N.-D. Nguyen, T. Do, T. D. Ngo, and D.-D. Le, "An evaluation of deep learning methods for small object detection," *Journal of Electrical and Computer Engineering*, vol. 2020, 2020.

[5] 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.

[6] M. Kisantal, Z. Wojna, J. Murawski, J. Naruniec, and K. Cho, "Augmentation for small object detection," *arXiv preprint arXiv:1902.07296*, 2019.

[7] F. Ozge Unel, B. O. Ozkalayci, and C. Cigla, "The power of tiling for small object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[8] P. Zhu, L. Wen, D. Du, X. Bian, Q. Hu, and H. Ling, "Vision meets drones: Past, present and future," *arXiv preprint arXiv:2001.06303*, 2020.

[9] X. Wu, D. Sahoo, and S. C. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, 2020.

[10] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *arXiv preprint arXiv:1905.05055*, 2019.

[11] 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*, Oct 2017.

[12] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *IEEE 12th International Conference on Computer Vision*, pp. 606–613, IEEE, 2009.

[13] H. Harzallah, F. Jurie, and C. Schmid, "Combining efficient object localization and image classification," in *IEEE 12th International Conference on Computer Vision*, pp. 237–244, IEEE, 2009.

[14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893, IEEE, 2005.

[15] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Proceedings of the International Conference on Image Processing*, vol. 1, pp. I–I, IEEE, 2002.

[16] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157, IEEE, 1999.

[17] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European Conference on Computer Vision*, pp. 404–417, Springer, 2006.

[18] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. I–I, IEEE, 2001.

[19] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, pp. 137–154, May 2004.

[20] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *Journal-Japanese Society For Artificial Intelligence*, vol. 14, no. 771-780, p. 1612, 1999.

[21] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, "Cascade object detection with deformable part models," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2241–2248, IEEE, 2010.

[22] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7310–7311, 2017.

[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, IEEE, 2009.

[24] S. Tang, M. Andriluka, and B. Schiele, "Detection and tracking of occluded people," *International Journal of Computer Vision*, vol. 110, no. 1, pp. 58–69, 2014.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.

[28] 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.

[29] 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.

[30] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[31] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1492–1500, 2017.

[32] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1251–1258, 2017.

[33] 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.

[34] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[35] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "Light-head R-CNN: In defense of two-stage object detector," *arXiv preprint arXiv:1711.07264*, 2017.

[36] 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, 2015.

[37] K. E. Van de Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders, "Segmentation as selective search for object recognition," in *International Conference on Computer Vision*, pp. 1879–1886, IEEE, 2011.

[38] 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.

[39] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, 2015.

[40] 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*, vol. 39, no. 6, pp. 1137–1149, 2016.

[41] 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.

[42] 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.

[43] 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*, pp. 779–788, 2016.

[44] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 761–769, 2016.

[45] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[46] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of the European Conference on Computer Vision*, pp. 734–750, 2018.

[47] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2110–2118, 2016.

[48] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1958–1970, 2008.

[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] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

[51] X. Yu, Y. Gong, N. Jiang, Q. Ye, and Z. Han, "Scale match for tiny person detection," in *the IEEE Winter Conference on Applications of Computer Vision*, pp. 1257–1265, 2020.

[52] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3974–3983, 2018.

[53] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5525–5533, 2016.

[54] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017.

[55] Z. Li and F. Zhou, "FSSD: Feature fusion single shot multibox detector," *arXiv preprint arXiv:1712.00960*, 2017.

[56] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[57] R. J. Wang, X. Li, and C. X. Ling, "Pelee: A real-time object detection system on mobile devices," *Advances in Neural Information Processing Systems*, vol. 31, pp. 1963–1972, 2018.

[58] Y. Liu, H. Li, J. Yan, F. Wei, X. Wang, and X. Tang, "Recurrent scale approximation for object detection in CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 571–579, 2017.

[59] B. Singh and L. S. Davis, "An analysis of scale invariance in object detection snip," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3578–3587, 2018.

[60] B. Singh, M. Najibi, and L. S. Davis, "Sniper: Efficient multi-scale training," in *Advances in Neural Information Processing Systems*, pp. 9310–9320, 2018.

[61] Z. Meng, X. Fan, X. Chen, M. Chen, and Y. Tong, "Detecting small signs from large images," in *IEEE International Conference on Information Reuse and Integration*, pp. 217–224, IEEE, 2017.

[62] Y. Pang, T. Wang, R. M. Anwer, F. S. Khan, and L. Shao, "Efficient featurized image pyramid network for single shot detector," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7336–7344, 2019.

[63] Z. Liu, G. Gao, L. Sun, and L. Fang, "IPG-Net: Image pyramid guidance network for small object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1026–1027, 2020.

[64] Y. Li, Y. Chen, N. Wang, and Z. Zhang, "Scale-aware trident networks for object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6054–6063, 2019.

[65] J. Nie, R. M. Anwer, H. Cholakkal, F. S. Khan, Y. Pang, and L. Shao, "Enriched feature guided refinement network for object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9537–9546, 2019.

[66] J. Jeong, H. Park, and N. Kwak, "Enhancement of SSD by concatenating feature maps for object detection," *arXiv preprint arXiv:1705.09587*, 2017.

[67] G. Cao, X. Xie, W. Yang, Q. Liao, G. Shi, and J. Wu, "Feature-fused SSD: fast detection for small objects," in *Ninth International Conference on Graphic and Image Processing*, vol. 10615, p. 106151E, International Society for Optics and Photonics, 2018.

[68] Q. Zhao, Y. Wang, T. Sheng, and Z. Tang, "Comprehensive feature enhancement module for single-shot object detector," in *Asian Conference on Computer Vision*, pp. 325–340, Springer, 2018.

[69] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, "M2DET: A single-shot object detector based on multi-level feature pyramid network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 9259–9266, 2019.

[70] Z. Du, J. Yin, and J. Yang, "Expanding receptive field YOLO for small object detection," in *Journal of Physics: Conference Series*, vol. 1314, p. 012202, IOP Publishing, 2019.

[71] M. Liu, X. Wang, A. Zhou, X. Fu, Y. Ma, and C. Piao, "UAV-YOLO: Small object detection on unmanned aerial vehicle perspective," *Sensors*, vol. 20, no. 8, p. 2238, 2020.

[72] C.-Y. Wang, H.-Y. Mark 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.

[73] 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.

[74] A. Rashwan, R. Agarwal, A. Kalra, and P. Poupart, "Matrixnets: A new scale and aspect ratio aware architecture for object detection," *arXiv preprint arXiv:2001.03194*, 2020.

[75] H. Lin, J. Zhou, Y. Gan, C.-M. Vong, and Q. Liu, "Novel up-scale feature aggregation for object detection in aerial images," *Neurocomputing*, vol. 411, pp. 364–374, 2020.

[76] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10781–10790, 2020.

[77] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "NAS-FPN: Learning scalable feature pyramid architecture for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7036–7045, 2019.

[78] Z. Liu, J. Du, F. Tian, and J. Wen, "MR-CNN: A multi-scale region-based convolutional neural network for small traffic sign recognition," *IEEE Access*, vol. 7, pp. 57120–57128, 2019.

[79] W. Liu, S. Liao, W. Hu, X. Liang, and Y. Zhang, "Improving tiny vehicle detection in complex scenes," in *IEEE International Conference on Multimedia and Expo*, pp. 1–6, IEEE, 2018.

[80] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 764–773, 2017.

[81] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," *arXiv preprint arXiv:1701.04128*, 2017.

[82] Z. Li, Y. Chen, G. Yu, and Y. Deng, "R-FCN++: Towards accurate region-based fully convolutional networks for object detection," in *Thirty-second AAAI Conference on Artificial Intelligence*, 2018.

[83] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2016.

[84] A. Torralba and P. Sinha, "Detecting faces in impoverished images," tech. rep., Massachusetts Inst of Tech Cambridge Artificial Intelligence Lab, 2001.

[85] Z. Song, Q. Chen, Z. Huang, Y. Hua, and S. Yan, "Contextualizing object detection and classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1585–1592, 2011.

[86] S. Gupta, B. Hariharan, and J. Malik, "Exploring person context and local scene context for object detection," *arXiv preprint arXiv:1511.08177*, 2015.

[87] X. Chen and A. Gupta, "Spatial memory for context reasoning in object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, Oct 2017.

[88] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert, "An empirical study of context in object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1271–1278, 2009.

[89] P. Hu and D. Ramanan, "Finding tiny faces," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1522–1530, 2017.

[90] V. Jain and E. Learned-miller, "FDDB: A benchmark for face detection in unconstrained settings," tech. rep., UMass Amherst Technical Report, 2010.

[91] C. Chen, M.-Y. Liu, O. Tuzel, and J. Xiao, "R-CNN for small object detection," in *Asian Conference on Computer Vision*, pp. 214–230, Springer, 2016.

[92] Y. Yuan, Z. Xiong, and Q. Wang, "VSSA-NET: Vertical spatial sequence attention network for traffic sign detection," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3423–3434, 2019.

[93] Y. Liu, S. Cao, P. Lasang, and S. Shen, "Modular lightweight network for road object detection using a feature fusion approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–13, 2019.

[94] W. Xiang, D. Zhang, H. Yu, and V. Athitsos, "Context-aware single-shot detector," in *IEEE Winter Conference on Applications of Computer Vision*, pp. 1784–1793, 2018.

[95] L. Guan, Y. Wu, and J. Zhao, "SCAN: Semantic context aware network for accurate small object detection," *International Journal of Computational Intelligence Systems*, vol. 11, pp. 951–961, 2018.

[96] Z. Ou, F. Xiao, B. Xiong, S. Shi, and M. Song, "FAMN: Feature aggregation multipath network for small traffic sign detection," *IEEE Access*, vol. 7, pp. 178798–178810, 2019.

[97] X. Hu, X. Xu, Y. Xiao, H. Chen, S. He, J. Qin, and P. Heng, "SINet: A scale-insensitive convolutional neural network for fast vehicle detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 1010–1019, 2019.

[98] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *Computer Vision – ECCV* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), (Cham), pp. 702–716, Springer International Publishing, 2016.

[99] M. F. Mathieu, J. J. Zhao, J. Zhao, A. Ramesh, P. Sprechmann, and Y. LeCun, "Disentangling factors of variation in deep representation using adversarial training," in *Advances in Neural Information Processing Systems* (D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, eds.), vol. 29, pp. 5040–5048, Curran Associates, Inc., 2016.

[100] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *Computer Vision – ECCV* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), (Cham), pp. 597–613, Springer International Publishing, 2016.

[101] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, July 2017.

[102] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, July 2017.

[103] Y. Bai, Y. Zhang, M. Ding, and B. Ghanem, "SOD-MTGAN: Small object detection via multi-task generative adversarial network," in *Proceedings of the European Conference on Computer Vision*, September 2018.

[104] Y. Chen, J. Li, Y. Niu, and J. He, "Small object detection networks based on classification-oriented super-resolution GAN for UAV aerial imagery," in *Chinese Control And Decision Conference*, pp. 4610–4615, 2019.

[105] J. Rabbi, N. Ray, M. Schubert, S. Chowdhury, and D. Chao, "Small-object detection in remote sensing images with end-to-end edge-enhanced GAN and object detector network," *Remote Sensing*, vol. 12, no. 9, p. 1432, 2020.

[106] Y. Bai, Y. Zhang, M. Ding, and B. Ghanem, "Finding tiny faces in the wild with generative adversarial network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[107] Y. Pang, J. Cao, J. Wang, and J. Han, "JCS-Net: Joint classification and super-resolution network for small-scale pedestrian detection in surveillance images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 12, pp. 3322–3331, 2019.

[108] H. Krishna and C. V. Jawahar, "Improving small object detection," in *4th IAPR Asian Conference on Pattern Recognition*, pp. 340–345, 2017.

[109] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," *arXiv preprint arXiv:1605.06409*, 2016.

[110] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "Reppoints: Point set representation for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, October 2019.

[111] C. Zhu, Y. He, and M. Savvides, "Feature selective anchor-free module for single-shot object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2019.

[112] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: A simple and strong anchor-free object detector," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[113] W. Ma, T. Tian, H. Xu, Y. Huang, and Z. Li, "AABO: Adaptive anchor box optimization for object detection via bayesian sub-sampling," in *Computer Vision – ECCV* (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), (Cham), pp. 560–575, Springer International Publishing, 2020.

[114] T. Yang, X. Zhang, Z. Li, W. Zhang, and J. Sun, "Metaanchor: Learning to detect objects with customized anchors," in *Advances in Neural Information Processing Systems* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), vol. 31, pp. 320–330, Curran Associates, Inc., 2018.

[115] Y. Zhong, J. Wang, J. Peng, and L. Zhang, "Anchor box optimization for object detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, March 2020.

[116] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *Computer Vision – ECCV* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), (Cham), pp. 75–91, Springer International Publishing, 2016.

[117] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Computer Vision – ECCV* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 391–405, Springer International Publishing, 2014.

[118] K.-K. Sung, "Learning and example selection for object and pattern detection," 1996.

[119] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin, "Region proposal by guided anchoring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2965–2974, 2019.

[120] P. Zhu, L. Wen, D. Du, X. Bian, H. Ling, Q. Hu, Q. Nie, H. Cheng, C. Liu, X. Liu, *et al.*, "VisDrone-DET2018: The vision meets drone object detection in image challenge results," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2018.

[121] D. Du, P. Zhu, L. Wen, X. Bian, H. Lin, Q. Hu, T. Peng, J. Zheng, X. Wang, Y. Zhang, *et al.*, "VisDrone-DET2019: The vision meets drone object detection in image challenge results," in *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 213–226, 2019.

Appendix A

Detailed Results of MS-COCO

Table A.1: MS-COCO Evaluation Results

Method	AP@0.5:0.95(all)	AP@0.5 (all)	AP@0.75 (all)	AP@0.5:0.95 (M)	AP@0.5:0.95 (L)
Faster R-CNN	24,2	45,3	23,5	26,4	37,1
Faster R-CNN	27,2	48,4	-	28,6	45
Faster R-CNN (FPN)	37,1	59,1	40,1	39,8	46,5
Faster R-CNN (FPN)	36,2	59,1	39	39	48,2
TridentNet	42,7	63,6	46,5	46,6	56,6
TridentNet (D)	46,8	67,6	51,5	51,2	60,5
TridentNet (MS)(D)	48,4	69,7	53,5	51,3	60,3
SNIPER (D)	47,6	68,5	53,4	50,6	60,7
ION	33,1	55,7	34,6	35,2	47,2
YOLOv2	21,6	44	19,2	22,4	35,5
YOLOv3	28,2	51,5	29,7	30,6	43,4
YOLOv3	31	55,3	32,3	33,2	42,8
YOLOv3	33	57,9	34,4	35,4	41,9
YOLOv3	36,2	60,6	38,2	37,4	46,1
YOLOv4	41,2	62,8	44,3	44,4	56
YOLOv4	43	64,9	46,5	46,1	55,2
YOLOv4	43,5	65,7	47,3	46,7	53,3
SSD	25,1	43,1	25,8	25,9	41,4
SDD	28,8	48,5	30,3	31,8	43,5
SSD	28	45,4	29,3	28,3	49,3
SDD	31,2	50,4	33,3	34,5	49,8

Table A.2: Continuation of Table A.1

Method	AP@0.5:0.95(all)	AP@0.5 (all)	AP@0.75 (all)	AP@0.5:0.95 (M)	AP@0.5:0.95 (L)
DSSD	28	46,1	29,2	28,1	47,6
DSSD	33,2	53,3	35,2	35,4	51,1
FSSD	27,1	47,7	27,8	29,2	42,2
FSSD	31,8	52,8	33,5	35,1	45
CFE-SSDv1	29,3	49	31	31,7	44,3
CFE-SSDv1	33,8	55,1	35,7	37,2	46,3
CFE-SSDv2	30,4	50,5	31,3	32,2	46,4
CFE-SSDv2	35,2	56,4	36,9	38,3	47,5
CFE-SSDv2	39,6	60,3	42,7	44,2	53,1
M2Det	33,5	52,4	35,6	37,6	47,6
M2Det	37,6	56,6	40,5	43,4	51,2
M2Det	34,3	53,5	36,5	38,8	47,9
M2Det	38,8	59,4	41,7	43,9	53,4
M2Det	41	59,7	45	46,5	53,8
EFGRNet	33,2	53,4	35,4	37,1	47,9
EFGRNet	37,5	58,8	40,4	41,6	49,4
EFGRNet	39	58,8	42,3	43,6	54,5
Pang et al[29] SSD (MS)	30	48,8	31,7	32,8	46,3
Pang et al[29] SSD (MS)	34,6	55,8	36,8	38,2	47,1
IPG RCNN	45,7	64,3	49,9	48,6	58,3
CornerNet511	40,6	56,4	43,2	42,8	54,3

Table A.3: Continuation of Table A.2

Method	AP@0.5:0.95(all)	AP@0.5 (all)	AP@0.75 (all)	AP@0.5:0.95 (M)	AP@0.5:0.95 (L)
MatrixNets	45,2	64,2	49,2	48,9	57,6
MatrixNets (MS)	47,8	66,2	52,3	50,4	60,7
FSAF	40,9	61,5	44	44,2	51,3
FSAF (MS)	42,8	63,1	46,5	45,5	53,2
FSAF	42,9	63,8	46,3	46,2	52,7
FSAF (MS)	44,6	65,2	48,6	47,1	54,6
FCOS*	47,9	65,9	52,5	50,7	59,7
FCOS	49	67,4	53,6	51,7	60,5
FCOS (D)	50,4	68,9	55	53	62,7
RetinaNet	32,5	50,9	34,8	35,8	46,7
RetinaNet	34,4	53,1	36,8	38,5	49,1
RetinaNet	35,7	55	38,5	38,9	46,3
RetinaNet	37,8	57,5	40,8	41,1	49,2
EfficientDet-D0	33,8	52,2	35,8	38,3	51,2
EfficientDet-D1	39,6	58,6	42,3	44,3	56
EfficientDet-D2	43	62,3	46,2	47	58,4
EfficientDet-D3	45,8	65	49,3	49,4	59,8
SOD-MTGAN	41,4	63,2	45,4	44,2	52,6

A COMPREHENSIVE SURVEY ON SMALL OBJECT DETECTION

ORIGINALITY REPORT

8%	4%	5%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	export.arxiv.org Internet Source	<1%
2	www.researchgate.net Internet Source	<1%
3	www.groundai.com Internet Source	<1%
4	Submitted to Nashville State Community College Student Paper	<1%
5	www.hindawi.com Internet Source	<1%
6	Submitted to Eastern Mediterranean University Student Paper	<1%
7	"Pattern Recognition and Computer Vision", Springer Science and Business Media LLC, 2018 Publication	<1%
	www.arehna.di.uoa.gr	