Automated Target Detection in Aerial Images using YOLOv3

Author

Wajiha Rahim Khan

Fall 2017-MS (CE) 00000203616

Supervisor

Dr. Muhammad Usman Akram

Co-Supervisor

Dr. Usman Qayyum (NESCOM)

DEPARTMENT OF COMPUTER ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

MARCH, 2020

Automated Target Detection in Aerial Images using YOLOv3

Author

Wajiha Rahim Khan

Fall 2017-MS (CE) 00000203616

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Computer Engineering

Thesis Supervisor:

Dr. Muhammad Usman Akram

Co-Supervisor

Dr. Usman Qayyum (NESCOM)

Thesis Supervisor's Signature: _____

DEPARTMENT OF COMPUTER ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

MARCH, 2020

Declaration

I certify that this research work titled "*Automated Target Detection in Aerial Images using YOLOv3"* is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Signature of Student

Wajiha Rahim Khan
Fall 2017-MS (CE) 00000203616

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature of Student

Wajiha Rahim Khan
MS (CE) 00000203616

Signature of Supervisor

Dr. Muhammad Usman Akram

Copyright Statement

- Copyright in text of this thesis rests with the student author. Copies (by any process) either in full, or of extracts, may be made only in accordance with instructions given by the author and lodged in the Library of NUST College of Electrical & Mechanical Engineering (CEME). Details may be obtained by the Librarian. This page must form part of any such copies made. Further copies (by any process) may not be made without the permission (in writing) of the author.
- The ownership of any intellectual property rights which may be described in this thesis is vested in NUST School of Mechanical & Manufacturing Engineering, subject to any prior agreement to the contrary, and may not be made available for use by third parties without the written permission of the CEME, which will prescribe the terms and conditions of any such agreement.
- Further information on the conditions under which disclosures and exploitation may take place is available from the Library of NUST School of Electrical & Engineering, Islamabad.

Acknowledgements

I am thankful to my Creator Allah Subhana-Watala to have guided me throughout this work at every step and for every new thought which You setup in my mind to improve it. Indeed I could have done nothing without Your priceless help and guidance. Whosoever helped me throughout the course of my thesis, whether my parents or any other individual was Your will, so indeed none be worthy of praise but You.

I would like to express my sincere gratitude to my advisor Dr. Muhammad Usman Akram for boosting my morale and for his continual assistance, motivation, dedication and invaluable guidance in my quest for knowledge. I am blessed to have such a co-operative advisor and kind mentor for my research.

Along with my advisor, I would like to acknowledge my entire thesis committee: Dr. Farhan Hussain, Dr. Arslan Shaukat from CEME and Dr. Usman Qayyum from NESCOM for their cooperation and prudent suggestions.

My acknowledgement would be incomplete without thanking the biggest source of my strength, my family. I am profusely thankful to my beloved parents who raised me when I was not capable of walking and continued to support me throughout every phase of my life and my loving siblings who were with me through my thick and thin.

This research work was funded by The National Engineering and Scientific Commission (NESCOM) under Agreement No. 00xx-RAC-IV/UNIV/2017.

Finally, I would like to express my gratitude to all my friends and the individuals who have encouraged and supported me through this entire period.

Dedicated to my exceptional parents and adored siblings whose tremendous support and cooperation led me to this wonderful accomplishment.

Abstract

In Computer vision, object detection and classification are active fields of research. Applications of object detection and classification includes a diverse range of fields such as surveillance, autonomous cars, robotic vision, search and rescue, driver assistance systems and military applications. In the last couple of decades, Convolution Neural Network (CNN) emerged as the most active field of research. There are a number of applications of CNN, and its architectures are used for the improvement of accuracy and efficiency in various fields. In this research, automated image interpretation for target detection and recognition in satellite/aerial images is presented that attracted much attention in past few years. But large images with complex background and the uneven distribution of trainings samples make it more challenging, particularly with small and dense objects. Recently various deep learning techniques mainly based on the CNN are proposed. The performance of all these techniques, however, depends on the situations they are use. However, in the context of object detection from satellite images we examine the performance of the latest CNN algorithms. This research details the procedure and parameters used for the training of convolutional neural networks (CNNs) on a set of aerial images for efficient and automated object detection. Potential application areas in the transportation and many other fields are also highlighted. The accuracy and reliability of CNNs depend on the network's training and the selection of operational parameters. The object detection results show that by selecting a proper set of parameters, a CNN can detect and classify objects with a high level of accuracy and computational efficiency. Furthermore, using a convolutional neural network implemented in the "YOLOv3" ("You Only Look Once") platform, we demonstrated that YOLOv3 not only exceeds in the sensitivity and processing time of other CNN algorithms but also in detecting small and dense targets. The effectiveness of the YOLOv3 framework has been demonstrated through extensive experiments and comprehensive evaluations on DOTA: A Large Scale Dataset for Object Detection in Aerial Images. This dataset contains high resolution images that are collected from the Google Earth, some are taken by satellite JL-1, and the others are taken by satellite GF-2 of the China Centre for Resources Satellite Data and Application.YOLOv3 achieves mAp of 61.94% that is 1.08% higher as compared to other detecting methods.

Key Words: *Target detection, DOTA, satellite images, YOLOv3, mAp.*

Table of Contents

List of Figures

List of Tables

Chapter 1: INTRODUCTION

In the present age of technology, Automatic Target Recognition & Detection is mainly a computer based technology related to computer vision and image processing that deals with recognizing & detecting instances of semantic targets of a certain class (such as humans, buildings, or cars) [1] in digital images and videos. One of the most common and well-researched domains of target recognition & detection include object detection i.e. face and pedestrian detection. Target detection has many applications mainly in many areas of computer vision, including image retrieval and video surveillance.

The problem of Target Detection in satellite imagery is a challenging one. Keeping in mind such challenges the problem of target detection from satellite/aerial images has been extensively studied from the past decades. Earlier we were unable to detect separate man-made or natural objects because of low resolution Satellite/Aerial Images. But with the availability of satellite and aerial images with high resolution (HRSI) (having sub meter resolution) we are able to recognize different range of objects and even can be separately identified than ever before which has opened new possibilities in the field of Automatic Detection of Targets in Satellite/Aerial Imagery. Considerable efforts have been made during the last decades to design and develop different algorithms and tools for target detection in satellite/aerial imagery like buildings, trees, roads, forests and vehicles. Different from previous studies this research focus on the use of deep learning techniques for target recognition & detection from satellite imagery emphasizing on detection of roads, buildings, solar panels, vehicles [2].

But here, our main focus is “Detection of Targets from Satellite or Aerial Imagery” and to figure out whether a satellite or aerial image has one or more targets(objects) that belong to the class of interest and if present locate their positions in the image. Satellite or aerial images contain different types of objects like vehicles, buildings, solar panels, roads, ships etc. Target Detection in Satellite Image analysis is also a fundamental problem which plays a significant role for different types of applications, such as detection of geological hazard, urban planning, Land use and cover mapping, environmental monitoring, updating of geographic information system and agriculture.

With the remarkable advancement in the quality and quantity of satellite images and due to the object appearance variations caused by illumination, shadow, background clutter, occlusion, viewpoint variation etc.

Object classification and detection is an active field of research since last couple of decades. There are many applications of object detection such as surveillance, autonomous vehicles and military purposes. Hand crafted features such as the Discrete Cosine Transform, Wavelet Transform and a few others are used for feature extraction and Principal Component Analysis (PCA) for classification [3]. In recent years CNN and deep learning architectures are used for efficient and accurate feature extraction as well as Classification and Segmentation.

In our target approach we are using data sets consisting of Satellite Images with different categories for training of the Convolution neural network. After training, rigorous tests are performed for validation of results**.** The algorithms used for Automatic Target Recognition/Detection work in different ways but the basic working principle of all these algorithms is same and that is Extraction of useful Features. They extract useful features from input data (Aerial/Satellite images in our case) and then on basis of these extracted features they detect and classify the required targets (that are of 15 different categories in our case).

In this research, we are using two Aerial images in from two different datasets along their augmentations. Aerial images are then passed through the localization and classification, network for localization as well as classification of objects.

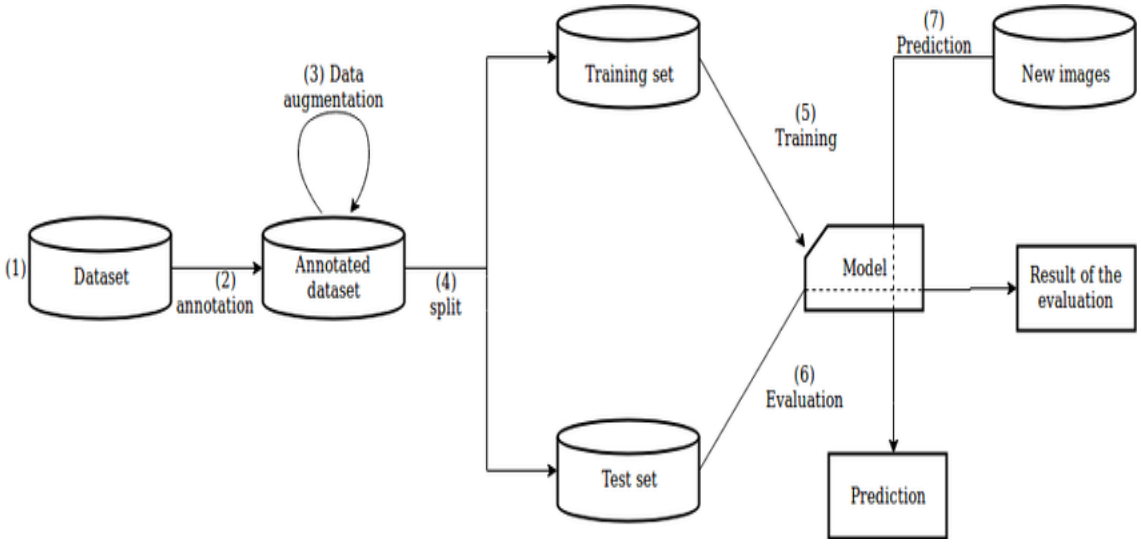

Figure 1.1: Workflow for Object Detection [1]

In the field of computer vision, object detection is still a crucial task and needs a lot of improvement to reach the level of human perception. The motivation behind this thesis was to utilize multiple modes of imaging to acquire a more informed image so that object detection becomes efficient. This system can be utilized in the war zone to detect stealth objects and can also be used in institutions for surveillance and security. Another prime purpose is to correctly detect the presence and accurately locate the object instance(s) in the image. It is (usually) a supervised learning problem in which, given a set of training images, one has to design an algorithm which can accurately locate and correctly classify as many object instances as possible in a rectangle box while avoiding false detections of background or multiple detections of the same instance. The images can have object instances from same classes, different classes or no instances at all.

1.1 Problem Statement

To develop a better solution for object detection and scene identification in Satellite/Aerial images so that it becomes easier and achievable for the purpose of safety and security.

1.2 Aims and Objectives

Major objectives of the research are as follows:

- It can be used in military be used in military e.g. for detecting and recognizing an object in war zones especially detecting unmanned aerial vehicles and cruise missiles of enemies.
- It can also be a used for keep the record of number of targets or simply counting targets, it is used for analyzing store performance or crowd statistics during festivals by counting people as targets.
- Can be used in vegetation for the detection of pesticides.

1.3 Contributions

- Review and Comparison of recent developments in object detection and localization systems using a convolutional neural network.

- Fully automated system for classification and localization of objects from aerial images for the purpose of safety and security.
- We trained and evaluated the model on DOTA dataset. The results show that our proposed network is really simple, fast and efficient. Both quantitative and qualitative comparisons of our network with the state-of-the-art networks are provided. system for object detection and recognition in satellite/aerial images using a convolutional neural network.

1.4 Structure of Thesis

This work is structured as follows:

Chapter 2: States the literature review of related research for image fusion techniques and object detection techniques.

Chapter 3: Gives the details about the dataset used in this thesis.

Chapter 4: Consists of the proposed methodology in detail. It includes: feature level object detection, decision level image classification, object detection and localization.

Chapter 5: Experiments and results are discussed in detail with all desired figures and tables.

Chapter 6: Concludes the thesis and reveals the future scope of this research.

Chapter 2: LITERATURE REVIEW

The task of automatically recognizing and locating objects in images and videos is important in order to make computers able to understand or interact with their surroundings. For humans, it is one of the primary tasks, in the paradigm of visual intelligence, in order to survive, work and communicate. If one wants machines to work for us or with us, they will need to make sense of their environment as good as humans or in some cases even better than humans. Solving the problem of object detection with all the challenges it presents has been identified as a major precursor to solving the problem of semantic understanding of the surrounding environment. A large number of academics as well as industry researchers have already shown their interest in it by focusing on applications, such as autonomous driving, surveillance, relief and rescue operations, deploying robots in factories, pedestrian and face detection, brand recognition, visual effects in images, digitizing texts, understanding aerial images, etc. which have object detection as a major challenge at their core [3]. Deep Learning has revolutionized the computing landscape leading to a fundamental change in how applications are being created. Applications are fast becoming intelligent and capable of performing complex tasks- tasks that were initially thought of being out of reach for a computer. Examples of these complex tasks include detecting and classifying objects in an image, summarizing large amounts of text, answering questions from a passage, generating art and defeating human players at complex games like Go and Chess. The human brain processes large amounts of data of varying patterns. It identifies these patterns, reasons about them and takes some action specific to that pattern. Artificial Intelligence aims to replicate this approach through Deep Learning. Deep Learning has proven to have been quite instrumental in understanding data of varying patterns at an accurate rate. This capability is responsible for most of the innovations in understanding language and images. With Deep Learning research moving forward at a fast pace, new discoveries and algorithms have led to disruption of numerous fields. One such field that has been affected by Deep Learning in a substantial way is object detection.

2.1 Object Detection

Object detection is the identification of an object in an image along with its localization and classification. Software systems that can perform these tasks are called object detectors. Object Detection has important applications. Numerous tasks which require human supervision can be automated with a software system that can detect objects in images. These include surveillance, disease identification and driving. The advent of deep learning has brought a profound change in how we implement computer vision nowadays. [4] Unfortunately, this technology has a high potential for irresponsible use. Military applications of object detectors are particularly worrying. Hence, in spite of its considerable useful applications, caution and responsible usage should always be kept in mind.

2.1.1 Object Detection Methods

The first object detector came out in 2001 and was called the Viola Jones Object Detector [5]. Although, it was technically classified as an object detector, its primary use case was for facial detection. It provided a real time solution and was adopted by many computer vision libraries at the time. The field was substantially accelerated with the advent of Deep Learning. The first Deep Learning object detector model was called the Over feat Network [6] which used Convolutional Neural Networks (CNNs) along with a sliding window approach. It classified each part of the image as an object/non object and subsequently combined the results to generate the final set of predictions. This method of using CNNs to solve detection led to new networks being introduced which pushed the state of the art even further. We shall explore these networks in the next section. There are currently two methods of constructing object detectors- the single step approach and the two step approach. The two step approach has achieved a better accuracy than the former whereas the single step approach has been faster and shown higher memory efficiency. The single step approach classifies objects in images along with their locations in a single step. The two step approach on the other hand divides this process into two steps. The first step generates a set of regions in the image that have a high probability of being an object. The second step then performs the final detection and classification of objects by taking these regions as input. These two steps are named the *Region*

Proposal Step and the *Object Detection Step* respectively. Alternatively, the single step approach combines these two steps to directly predict the class probabilities and object locations.

Object detector models have gone through various changes throughout the years since 2012. The first breakthrough in object detection was the RCNN [6] which resulted in an improvement of nearly 30% over the previous state of the art. We shall start the survey by exploring this detector first.

2.1.1.1 Two Stage Detectors

Region Convolutional Network (R-CNN)

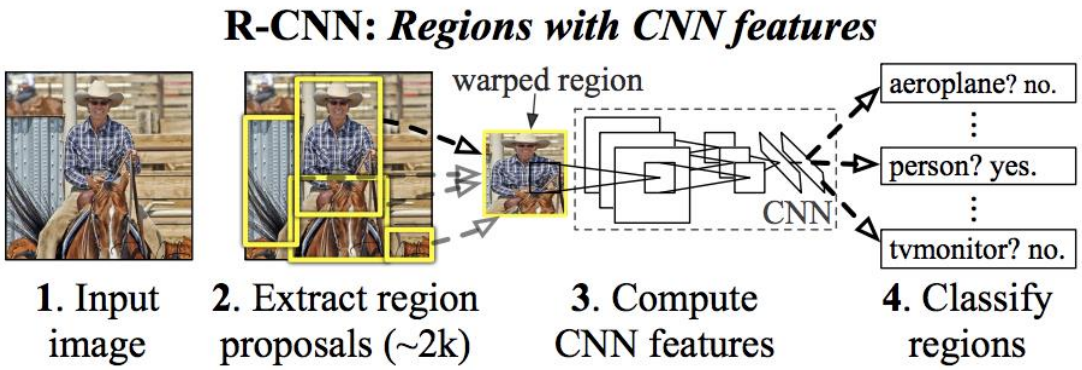

Figure. 2.1: Region Convolutional Network [6]

The RCNN [6] Model was a highly influential model that has shaped the structure of modern object detectors. It was the first detector which proposed the two step approach. We shall first look at the Region Proposal Model now.

Fast RCNN

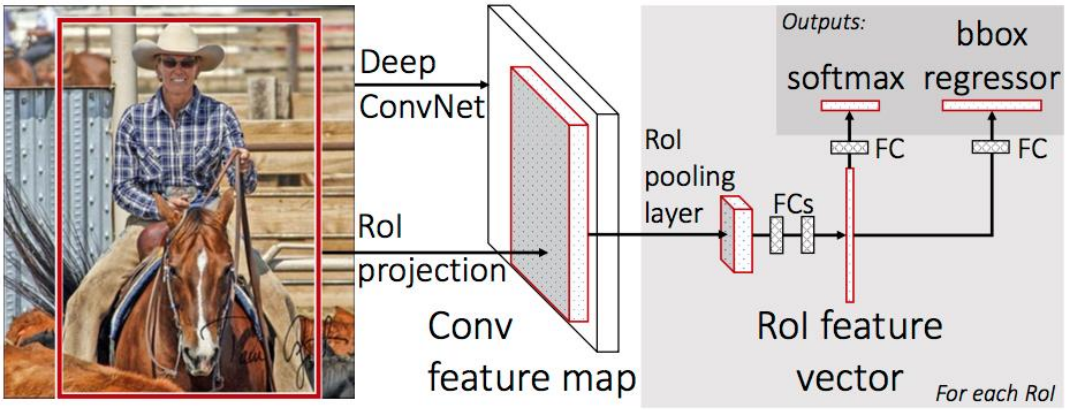

Figure. 2.2: Fast Region Convolutional Network [7]

The Fast RCNN [7] came out soon after the RCNN and was a substantial improvement upon the original. The Fast RCNN is also a two-step model which is quite similar to the RCNN, in that it uses selective search to find some regions and then runs each region through the object detector network. This network consists of a convolutional base and two SVM heads for classification and regression. Predictions are made for the class and offsets of each region. The RCNN Model takes every region proposal and runs them through the convolutional base. This is quite inefficient as an overhead of running a region proposal through the convolutional base is added, every time a region proposal is processed. The Fast RCNN aims to reduce this overhead by running the convolutional base just once. It runs the convolutional base over the entire image to generate a feature map. The regions are cropped from this feature map instead of the input image. Hence, features are shared leading to a reduction in both space and time. This cropping procedure is done using a new algorithm called ROI Pooling.

Faster RCNN

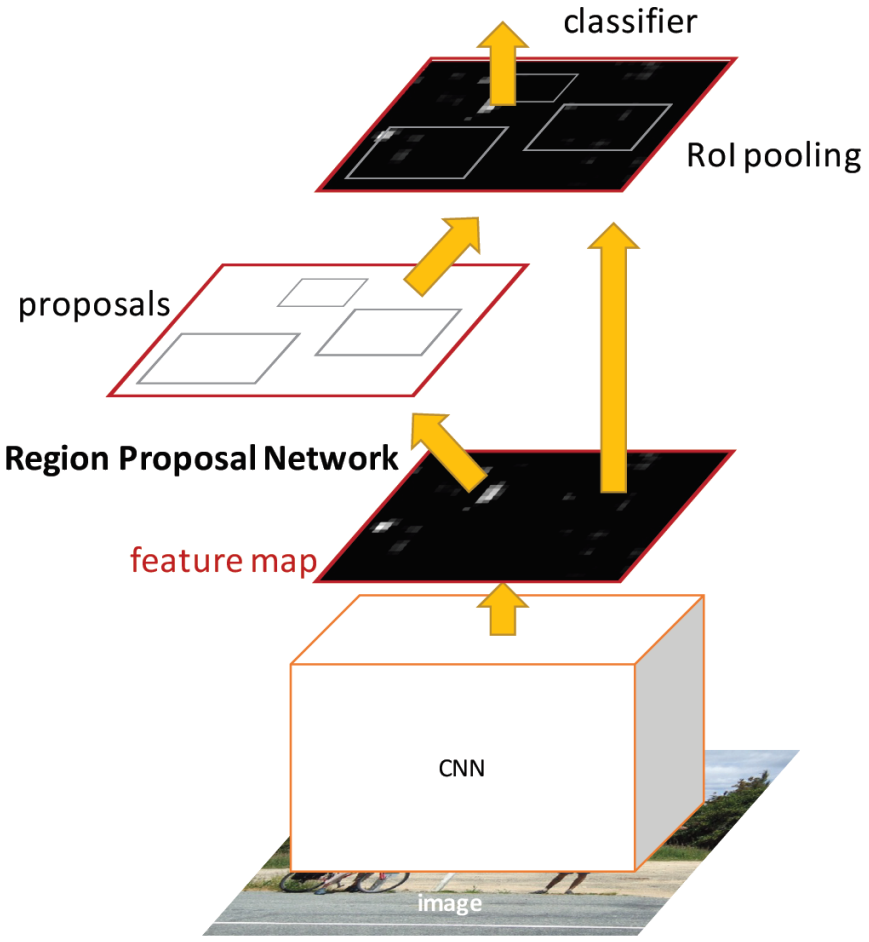

Figure. 2.3: Faster RCNN [8]

The Faster RCNN [8] came out soon after the Fast RCNN paper. It was meant to represent the final stage of what the RCNN set out to do. It proposed a detector that was learnt end to end. This entailed doing away with the algorithmic region proposal selection method and constructing a network that learned to predict good region proposals. Selective Search was serviceable but took a lot of time and set a bottleneck for accuracy. A network that learnt to predict higher quality regions would theoretically have higher quality predictions.

The Faster RCNN introduced the *Region Proposal Network* (RPN) to replace Selective Search. The RPN needed to have the capability of predicting regions of

multiple scales and aspect ratios across the image. This was achieved using a novel concept of anchors.

TABLE 2.1: Comparison Table of Two Stage Object Detection Methods

Object Detector Type	Backbone	AP	AP50	AP75	APS	APM	APL
Faster R-CNN+++[6]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [6]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [6]	Inception-ResNet-v2 [3]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [6]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	52.1

2.1.1.2 Single Step Object Detectors

Single Step Object Detectors have been popular for some time now. Their simplicity and speed coupled with reasonable accuracy have been powerful reasons for their popularity. Single step detectors are similar to the RPN network, however instead of predicting objects/non objects they directly predict object classes and coordinate offsets [9].

Single Shot Multi Box Detector (SSD)

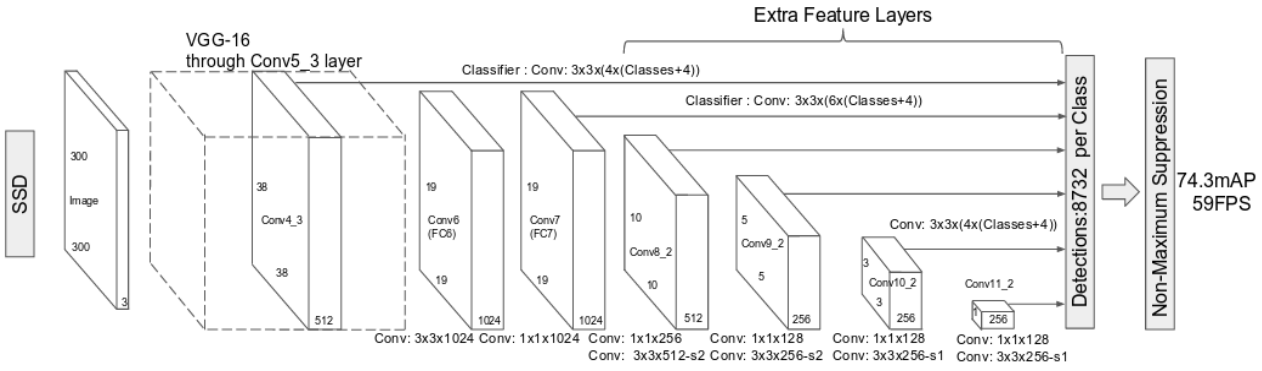

Figure. 2.6: Single Shot Multi Box Detector (SSD) [9]

Single Shot Multi Box Detector [10] came out in 2015, boasting state of the art results at the time and real time speeds. The SSD uses anchors to define the number of default regions in an image. As explained before, these anchors predict the class scores and the box coordinates offsets. A backbone convolutional base (VGG16) is used and a multi task loss is computed to train the network. This loss is similar to the Faster RCNN loss function- a smooth L1 loss to predict the box offsets is used along with the cross entropy loss to train for the class probabilities [10]. The major difference between the SSD from other architectures is that it was the first model to propose training on a feature pyramid.

The network is trained on n number of feature maps, instead of just one. These feature maps, taken from each layer are similar to the FPN network but with one important difference. They do not use top down pathways to enrich the feature map with higher level information. A feature map is taken from each scale and a loss is computed and back propagated. Studies have shown that the top down pathway is important in ablation studies. Modern object detectors modify the original SSD architecture by replacing the SSD feature pyramid with the FPN. The SSD network computes the anchors for each scale in a unique way. The network uses a concept of aspect ratios and scales, each cell on the feature map generates 6 types of anchors, similar to the Faster RCNN. These anchors vary in aspect ratio and the scale is captured by the multiple feature maps, in a similar fashion as the FPN. SSD uses this feature pyramid to achieve a high accuracy, while remaining the fastest detector on the market. Its variants are used in production systems today, where there is a need for fast low memory object detectors. Recently, a tweak to the SSD architecture was introduced which further improves on the memory consumption and speed of the model without sacrificing on accuracy. The new network is called the Pyramid Pooling Network [10]. The PPN replaces the convolution layers needed to compute feature maps with max pooling layers which are faster to compute.

You Only Look Once (YOLO)

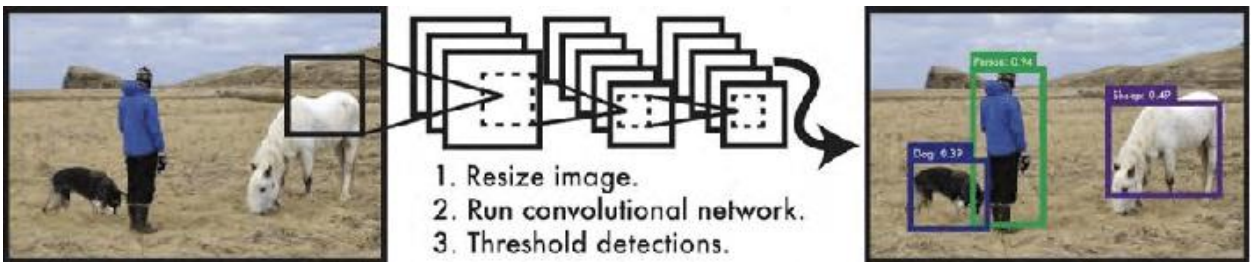

Figure. 2.7: You Only Look Once (YOLO) [11]

The YOLO [11] group of architectures were constructed in the same vein as the SSD architectures. The image was run through a few convolutional layers to construct a feature map. The concept of anchors was used here too, with every grid cell acting as a pixel point on the original image. The YOLO algorithm generated 2 anchors for each grid cell. Unlike the Fast RCNN, Yolo has only one head. The head outputs feature map of size 7 by 7 by (x+1+5∗ (k)), k is the number of anchors, x+1 is the total number of classes including the background class. The number 5 comes from the four offsets of x, y, height, width and an extra parameter that detects if the region contains an object or not. YOLO coins it as the object ness of the anchor.

Retina Net

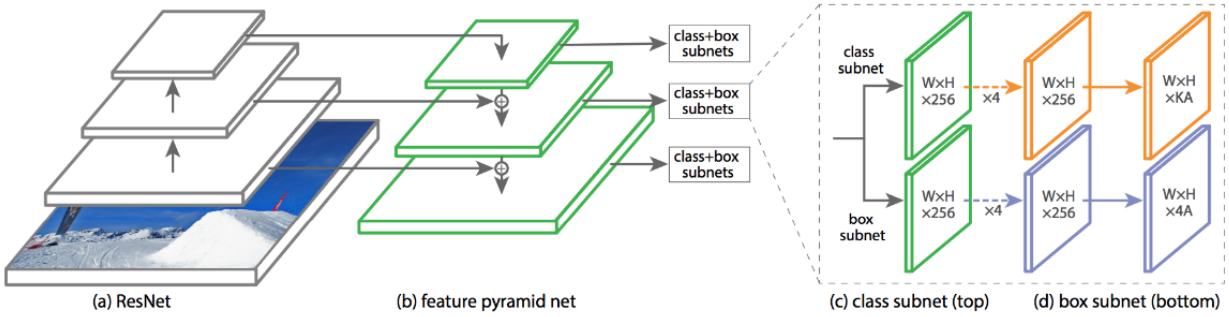

Figure. 2.8: Retina Net [12]

The Retina Net is a single step object detector which boasts the state of the art results at this point in time by introducing a novel loss function [12]. This model represents the first instance where one step detectors have surpassed two step detectors in accuracy while retaining superior speed.

The authors realized that the reason why one step detectors have lagged behind 2 step detectors in accuracy was an implicit class imbalance problem that was encountered while training. The Retina Net sought to solve this problem by introducing a loss function coined Focal Loss.

TABLE 2.2: Comparison Table of Single Stage Object Detection Techniques

Object Detector Type	Backbone	AP	AP50	AP75	APS	APM	APL
YOLOv2 [11]	DarkNet-19 [11]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [5, 4]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [18, 4]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
Retina Net[16, 4]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
Retina Net[16, 17]	ResNeXt-101-FPN	40.8	61.1	44.1	24.1	44.2	51.2

2.2 Convolutional Bases

All modern object detectors have a convolutional base. This base is responsible for creating a feature map that is embedded with salient information about the image. The accuracy for the object detector is highly related to how well the convolutional base can capture meaningful information about the image [13]. The base takes the image through a series of convolutions that make the image smaller and deeper. This process allows the network to make sense of the various shapes in the image.

Convolutional networks form the backbone of most modern computer vision models. A lot of convolutional networks with different architectures have come out in the past few years. They are roughly judged on three factors namely accuracy, speed and memory.

Convolutional bases are selected according to the use case. For example, object detectors on the phone will require the base to be small and fast. Alternatively, larger bases will be used by the powerful GPU's on the cloud. A lot of research has gone

into making these convolutional nets faster and more accurate. A few popular bases are described in the coming section. Bigger nets have led in accuracy however, advancements have been made to compress and optimize neural networks with a minimal tradeoff on accuracy.

Convolutional Neural Networks are acting as a backbone solution for computer vision and machine learning tasks these days along with their growing popularity with every passing day. CNNs are widely admired and used due to their simple and understandable architecture as well as their potential to provide efficient solutions. First ever proposed architecture for Convolutional neural network was LeNet and it was published by LeCun et al [14]. It takes an input image perform convolution on it using 5x5 filters with a stride of 1, then performs sub sampling after that some more convolutions and few pooling layers and then few fully connected layers. This architecture provided excellent results for digit recognition. This architecture provided the basis for today's fast and efficient networks, but it could not provide better results on data sets due lack of availability of data on the internet and due to less computational power of GPUs present at that time.

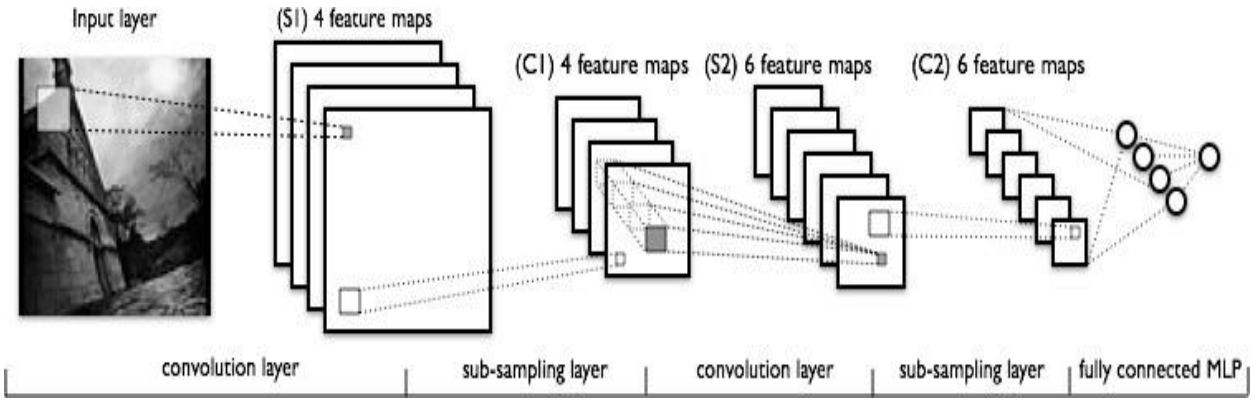

Figure 2.1: LeNet Architecture [14]

In 2012 Alex Krizhevsky et al. [15] proposed a CNN architecture called AlexNet. AlexNet won 2012 ImageNet classification challenge and beat up all other methods by a clear margin. It reduced error rate to 16.4%. Its architecture consists of five Convolutional layers, three Max pooling layers, two Normalization layers and three fully connected layers.

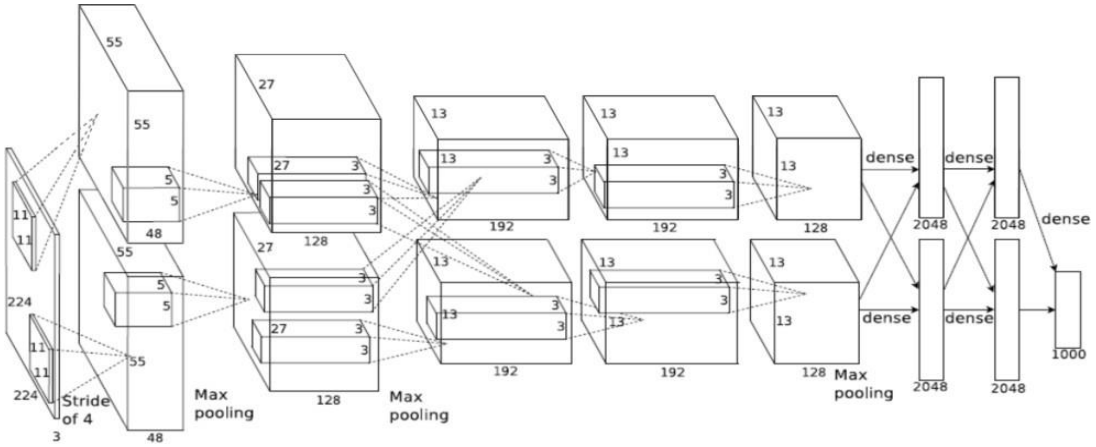

Figure 2.2: AlexNet Architecture [15]

With time CNNs became deeper by increasing layers, in AlexNet there were eight layers in total while in 2014 two networks emerged named as VGGNet [16] and GoogleNet. In VGGNet there were nineteen layers with smaller filters. Convolution layer filters were 3x3 with stride 1 and pad 1. Max pooling layers were 2x2 with stride 2. 7.3% error rate in ILSVR challenge.

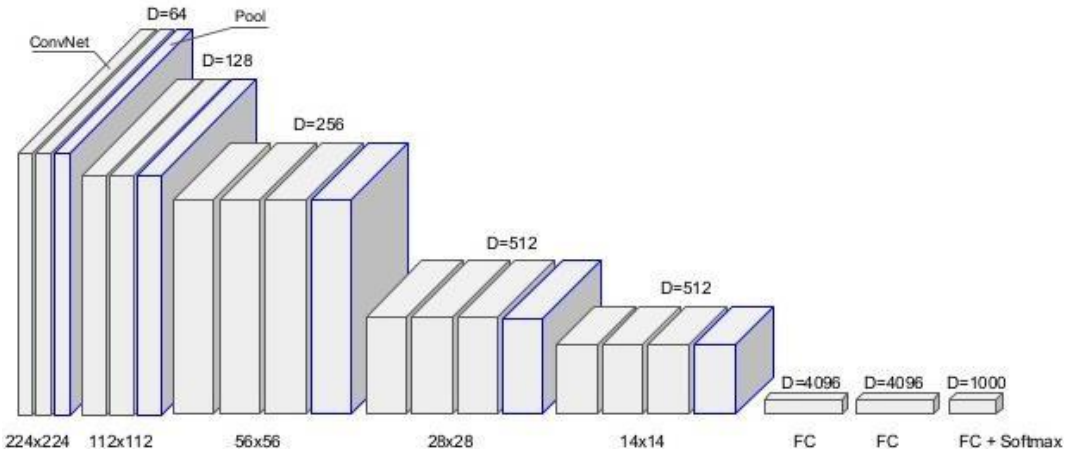

Figure 2.3: VGG-Net Architecture [16]

GoogleNet [17] won the classification challenge of ILSVR in 2014. In GoogleNet there were twenty two layers. GoogleNet used inception modules and is formed by placing multiple inception modules above each other. Error rate decreased to 6.7%. In the inception module, on each input coming from previous module multiple convolution

are applied along with a pooling layer and results of all of them are then combined in a single layer which will then be fed to next module and in this way it goes on. In GoogleNet at the start of the network, we have stem network having convolution and pooling layers to start the network operations. Then we have inception modules placed above each other. After these inception modules we have output module also called classifier module for output classification. There are no fully connected layers in this architecture. Major problem of this architecture was computational complexity as each inception module will increase the depth of the output.

In 2015 ResNet [18] won ILSVR challenge it turned out to be the most dense architecture. ResNet has 152 layers in total. An Error rate of ResNet decreased to 3.57%. It is considered to be the best CNN architecture by far and it is used widely with small modifications here and there.

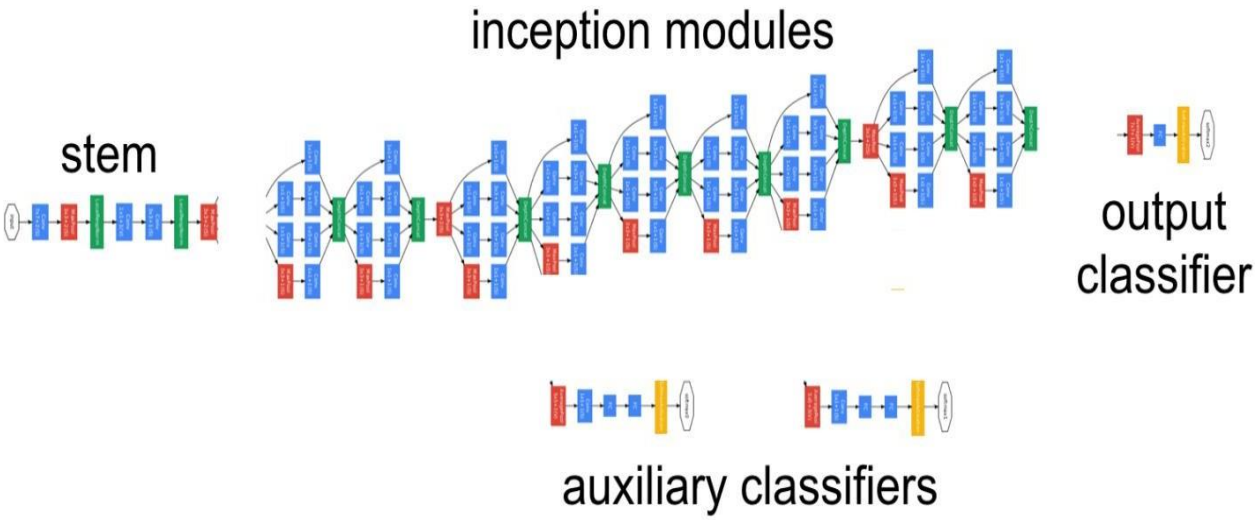

Figure 2.5: ResNet Residual block [18]

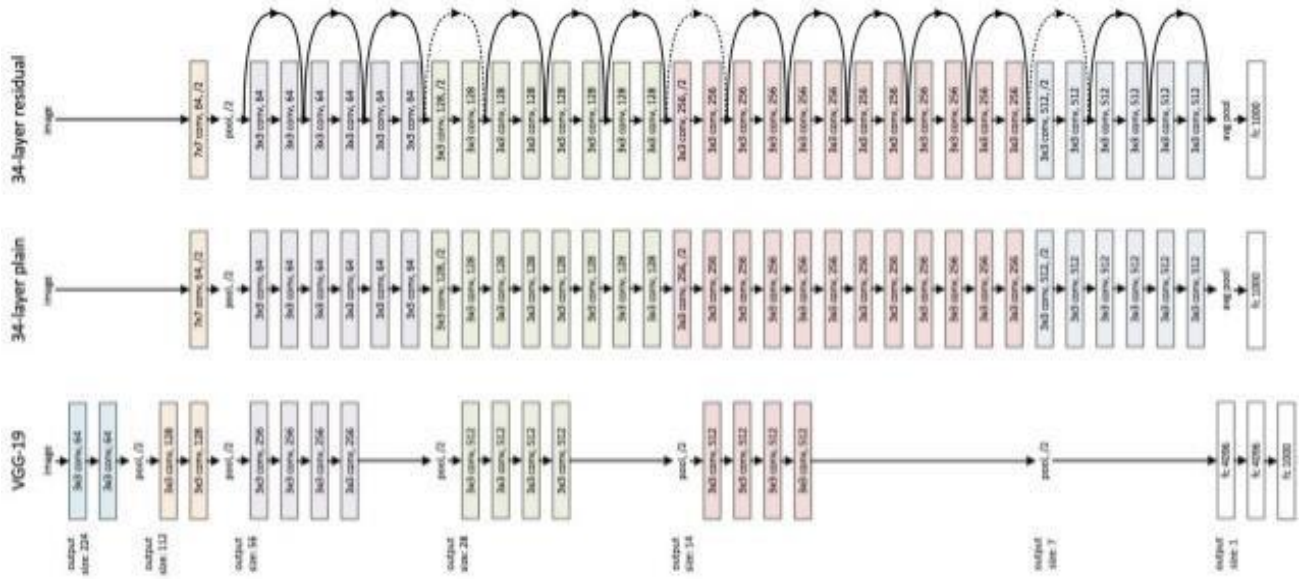

Figure 2.6: ResNet Architecture [18]

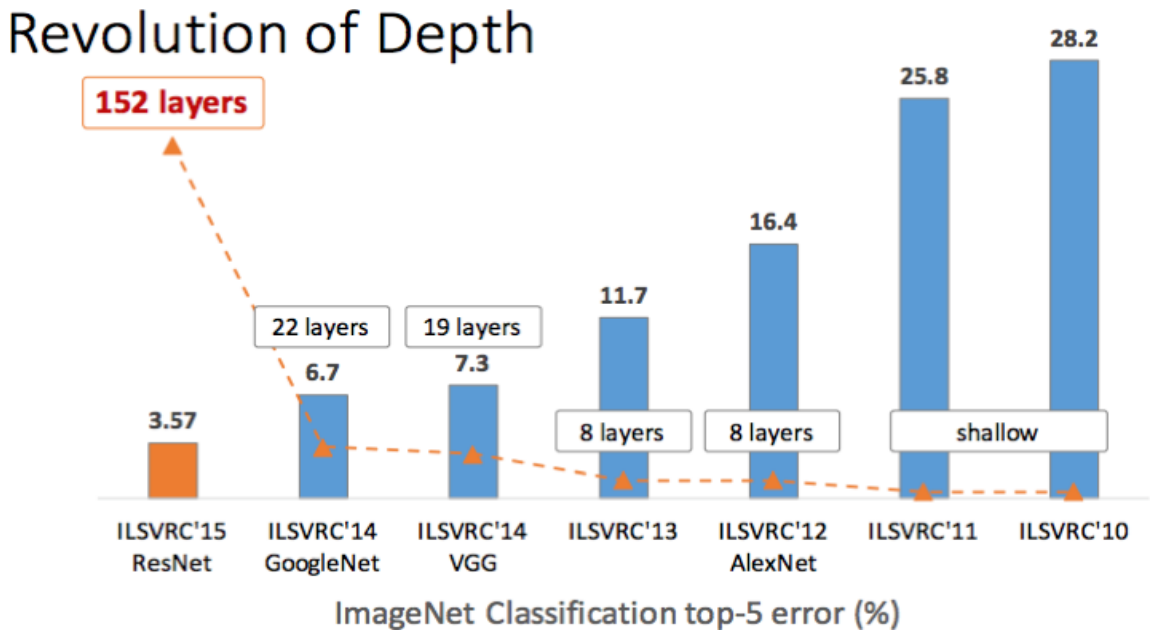

Figure 2.7: Comparison of Error rates of CNNs [20]

Table 2.3: Comparison of CNN Architectures

CNN	Conv.layers	MACCs (millions)	Parameters (millions)	Activations (millions)	ImageNet Top-5 error

AlexNet	5	1140	62.4	2.4	16.4%
VGGNet-16	16	15470	138.3	29.0	8.1%
GoogleNet	22	1600	7.0	10.4	6.7%
ResNet-50	50	3870	25.6	46.9	3.5%

2.3 Applications of CNN

The Convolutional Neural Network has emerged as the best possible technique for detection and classification in the last couple of decades. After successful classification of ImageNet dataset, CNN was opted to classify PASCAL VOC and ILSVR2013 data set challenge [20]. Ross Girshick et al [21]. From UC Berkley accepted this challenge and proved that CNN is a far better approach than HOG and SIFT for classification as well as segmentation of images. They used region based convolutional neural network for object detection and named it R-CNN. Their technique involved: taking images as input, extracting regions from the data set, computation of features for each region using convolutional neural network and then the classification of a region using SVM. [22]

Mathias Limmer et al. [23] Proposed another useful technique for colorization of images, which is used to transfer RGB images to near infrared images using the convolutional neural network. Basic image colorization techniques comprise of three steps: first step is to segment images, the second is to assign a color palette for each region and in third step each palette is used to determine chrominance. During preprocessing step image pyramid is created than pyramid's levels are assigned to the corresponding layer of the neural network. At the end of the neural network all layers are fused together to form on a fully connected layer of output. Size of filter changes after each pooling layer while the size of kernel remains same throughout the network. Once CNN's result is obtained post processing is done to remove noise and ambiguities. For

post processing, bilateral filters are used and once image is filtered it is compared with input image for high frequency augmentation to form a resultant colorized image.

CNN can be used in the field of medicine as well. In [24] Teresa Araujo et al. proposed a CNN based technique which helps in the detection of cancerous cells in breast tissues. Deep neural networks are found to be more effective than other conventional techniques. CNN's output provides certain features of the image which can then be combined with SVM later to get even better results.

Gustav Larsson et al. Proposed a technique based on the deep CNN for colorization of grayscale images. As colored images provide much more information than black and white or grayscale images, it is of great consideration to convert already present grayscale datasets into colored datasets. The Convolutional neural networks can be used as a major and effective technique for this purpose as already trained networks produce efficient results. Proposed technique takes a grayscale image as input and produce colorized output by passing it through some Convolutional layers and couple of fully connected layers. [25]

Another application of convolutional neural network is to check out the potential difference between colored and LiDAR data for object detection and classification. R. Niessner et al. discussed this problem in detail and provided experimental conclusions about better techniques and effective approach. Three different approaches are taken into account for the variation of convolutional neural network. In the first method a Convolutional neural network is taken which is trained already and its output is given as an input into SVM for classification, In the second technique another pre trained neural network is taken and is used for the purpose of refinement of data within the network layers, and the third an last approach is to take a neural network which is not trained in advance and use it for the purpose of classification after training it. GRSS data provided by IEEE are used in the verification process. Upon evaluation it is concluded that LiDAR data provide better results as compared to RGB [26].

Image matting is a popular technique in imaging and video editing areas. This could be very useful when dealing with background alteration and film making, especially in animated or movies with special effects. To crop an object or number of objects from the foreground with such accuracy that on adding a virtual background it does not look

unrealistic. Ning Xu et al. Proposed image matting technique based on neural networks, which also handles high level features and context of images. Their architecture consists of two neural networks, one of which takes the input image and its tri-map and uses encoder decoder mechanism to produce matte of the image. Second network is used for fine tuning of output of the first network. [27]

In addition to above mentioned techniques convolutional neural network can be used in a number of other applications as well. In [28] Ashnil Kumar et al. combined multiple Convolutional neural networks to ensemble data from different modalities which can be used for medical image classification and detection. In [29] Ronald Kemker et al. used deep CNN for the purpose of image segmentation. Multi spectral images are being segmented in this method and networks are trained and fine-tuned over multi spectral datasets. Jiwen Lu et al. [30] proposed a new technique for image classification based on multi manifold deep metric. They utilized manifold models for classification of objects under certain circumstances, such as different lighting conditions and different angled images of the same scene. Simon Philipp Hohberg [31] did his thesis on the topic of wildfire smoke detection by using convolutional neural networks, which is a critical topic as detection of wildfire smoke at early stages can be very helpful in controlling the fire that's spreading in the area and hence saving a lot of resources, wildlife and human lives. In [32] Jin Kyu Kang et al. Proposed a fuzzy inference based Convolutional neural network, which takes RGB and infrared images as inputs and process them to find out which one is giving better results and in turn perform pedestrian detection. In [33] Natalia Neverova used convolutional neural network for human motion analysis. It takes inputs in different forms such as images, videos, audio and recorded voice and then combine the results of these modalities to give an estimation of human's emotional and physical state. Samer Hijazi et al. Used Convolutional neural network for image recognition. [34]

2.4 Automatic Target Recognition & Detection (ATR)

Current approaches to image classification make essential use of artificial intelligence approaches. The introduction of deep learning has revolutionized the artificial intelligence and computer vision field. In this research, we will explore innovative deep

learning approaches for vehicle classification and localization using real traffic surveillance recordings. As we all know humans interpret an image as a meaningful arrangement of regions and objects not just a random collection of pixels. There exist a large variety of images like: natural scenes, paintings, aerial/satellite images etc [1]. Humans have no problem to interpret and classify these images despite the large variations in these images. Scene understanding and classification is a challenging problem in image analysis. Classification basically works on the concept of isolation like based on some specific criteria separating the different regions, which generally correspond to meaningful objects that is here to compose or we can say to complete a given scene. Deep neural networks especially convolutional neural network will be explored in this research. The proposed image classification approach can be used for automatic target recognition for general scene understanding. We develop a Convolution Neural Networks (CNN) architecture for achieving ATR in aerial imagery that classify and localize the vehicles and pedestrians and find what will be the classification accuracy levels that can be achieved through the application of neural networks. We use graphics processing units (GPU) to accomplish the computational tasks.

Applications of Automatic Target Recognition & Detection

Automatic Target Recognition and Detection has multiple applications and can be used in a variety of fields mainly depends on the targets need to be recognize or detect like:

- **Facial Recognition**

This is basically an application of ATR i.e. called the “Deep Face” that has been developed by a group of researchers in the Facebook and is used to identify human faces in a digital image. Facial Recognition is basically based on classification and detection of various components of face like the eyes, nose and etc.

- **Target Counting**

ATR can be also used for keep the record of number of targets or simply counting targets, it is used for analyzing store performance or crowd statistics during festivals by counting people as targets [2]. It is a very important application of ATR and can be used in multiple fields.

- **Quality Check in Industries**

ATR is also used to identify and classify different products in industry. Finding a product of a specific category through visual inspection is a basic task that is involved in multiple industrial processes like sorting, inventory management, machining, quality management, packaging etc.

- **Self-Driving Cars**

As we all know Self-driving cars are the Future and there's no doubt in that. But the working mechanism behind it is very difficult and completely based on ATR as it combines a variety of techniques to perceive their surroundings, including radar, laser light, GPS and computer vision [3].

- **Security**

ATR plays a very important role in Security. Be it face ID of Apple or the retina scan used in all the sci-fi movies. It is also used by the government to access the security feed and match it with their existing database to find any criminals or to detect the robbers' vehicle. Similarly ATR can be used in military e.g. for detecting and recognizing an object on a battlefield especially detecting unmanned aerial vehicles and cruise missiles of enemies.

ATR has applications in many areas of computer vision, including image retrieval and surveillance and video object co-segmentation. Most important application is in intelligent traffic surveillance systems to analyze and extract useful information from recordings [2]. ATR can be used to identify and classify manmade objects as well as for biological targets such as animals, humans etc. It is also used in tracking objects, for example tracking a ball during a football match, tracking movement of a cricket bat, tracking a person in a video.

The applications of Automatic Target Recognition and Detection are limitless.

2.5 Automated Target Detection in Satellite/Aerial Images using CNN

Today, Recognition & Detection of Targets in Satellite/Aerial Images is one of the most challenging problems. However, due to importance of ATR in a wide range of applications such as military applications , urban planning , and environmental management it has attracted a lot of researcher's attention in recent years and is

considered as an essential step for understanding and interpreting large Aerial/Satellite scenes [1] . Thus, researchers have proposed different techniques and algorithms in order to accurately recognize and detect different types of targets in Satellite/Aerial Images such as vehicle, airplane, buildings, and storage tanks etc.

The techniques that have been proposed in the literature for solving Target recognition and detection task in Satellite/ Aerial Imagery can be classified into two main categories: traditional approaches that rely on handcrafted features and deep learning-based approaches that rely on a convolution neural network (CNN) as feature extractor and provide superior performance. Handcrafted features limit the representation capacity and do not give the desired accuracy. On the other hand, deep learning shows an outstanding performance in many domains such as image processing due to automatic features generation.

The development of intelligent traffic surveillance systems has emerged as an important issue in recent years. One of the major issues faced by today's imagery analyst (IA) community is the timely exploitation of large quantities of imagery. Extensive search areas, multi-banded image cubes, and high data rates create data volumes that are often too large to be analyzed within operational timeline requirements. So an ATR system can overcome this difficulty along with correct detection and classification of targets within an image/scene.

A lot of techniques have been proposed in early 90s regarding object detection and classification [36]. Some of them include use of image information along with classifiers to divide the problem into certain categories like image preprocessing, feature extraction, path planning and object detection etc. So the automated target recognition would be broken up into sub-categories and then individual implementation would be done.CNNs are recently being used for automated targets to develop an end to end system that can detect and classify targets accurately just like humans but there is still a need for improvement to attain best possible results. The area of Detection and classification in natural imagery is where most ATR related neural network research has been accomplished.

Li Zhuo1 and Liying Jiang1 presented a new method [37] of classification using Convolutional Neural Networks for vehicles .the presented method used the concept of

per training and fine-tuning .initial model is trained on a subset of ImageNet dataset ,then this model is fine tuned for the constructed vehicle dataset. The constructed dataset has total of almost 13.7 thousand images with six categories of vehicles that are extracted from different traffic videos. The experimental results obtained from this are 3% higher than the other common methods used for classification of vehicles from natural images with an accuracy of 98.26%.Yohei Koga, Hiroyuki Miyazaki and Ryosuke Shibasaki [38] proposed a hard mining CNN based method for detecting vehicles in aerial images. Basically hard mining is applied to choose most informative data for training of CNN.This approach make best Training of CNN by providing the aerial images that only contain the vehicles no other category was present in those images. This method is used for efficient training of a network. The results obtained after the training of CNN on the hard mined data are .02% higher than the other methods that train CNN through conventional training method. Jun Sang , Zhengyuan Wua proposed a new model [39] for vehicle detection known as YOLOv2.The proposed model used the famous k mean clustering algorithm for bounding boxes of training dataset. Similarly fusion strategy with the concept of multi-layer features was used in this model in order to improve the ability of a network to extract features. Validation of this model is carried on a Beijing Institute of Technology (BIT) dataset which has 9.5 thousand images. The validation results with the accuracy point of 94.78% of this model indicates that it is one of the best models for detecting vehicles from the images. In recent years, regional based DNNs especially CNN gain a lot of popularity in the case of target detection in many computer vision applications .however their use as detection algorithm in images has many shortcomings like they cannot accurately detect the small targets like small vehicles and similarly they cannot distinguish targets and complex backgrounds.in order to overcome these and many other shortcomings of regional based convolutional networks Tianyu Tang and Shilin Zhou [40] proposed a new and improved method of detecting targets based on faster R-CNN.They basically extract targets like vehicles through hyper region proposal network and then classify the extracted targets through cascade boosted classifiers in aerial images. This is an innovative method used for target detection. This method is tested over different datasets of aerial images and the

experimental results of proposed method are quite satisfying as compared to other methods.

Table 2.4: Literature Review Summary

S. No.	Author	Year	Technique	Data set	Reported results
1	Ross Girshick et al.	2014	CNN to bottom-up region proposals	PASCAL VOC	53.3% mAP
2	Abdullah Asım YILMAZ	2018	R-CNN, Faster R-CNN	sample vehicle datasets	0.73, 0.76 mAP
3	Yohei Koga	2018	HEM to SGD on data to train CNN	Images from New York	0.02 F1 Score
4	Yi Tan	2017	vehicle proposals and CNN	infrared (IR) data	85% accuracy
5	Heikki Huttunen	2016	DNN and SVM using SIFTfeatures	database 6500 images	97 % accuracy
6	Jorg Wagner et al	2016	CNN	KAIST	43.80% miss rate
7	Jingjing Liu t al	2016	Faster-RCNN	KAIST	37% miss rate
8	Zhen Dong	2015	CNN with Laplacian filtert softmax classifier	BIT-Vehicle dataset,	88.1% accuracy
9	Jun Sang	2019	YOLOv2_Vehicle based on YOLOv2	BIT-Vehicle validation dataset	94.78% mAp

This table concludes a general overview of the literature based on the mainstream object detection halfway through 2014. Although the methods presented are all different, it has been shown that in fact most papers have converged towards the same crucial design choices. All pipelines are now fully convolutional, which brings structure

(regularization), simplicity, speed and elegance to the detectors. The need to use multi-scale information from different layers of the CNN is now apparent. Based on this, most of the research being done now in the mainstream object recognition consists of inventing new ways of passing the information through the different layers or coming up with different kinds of losses or parametrization. Based on these developments, we presented our approach in the mainstream object detection in satellite images in chapter 4 .

Chapter 3: DATASET

In the field of CNN Data is one of the main challenges and major requirements in order to attain considerable results. Le Cun et al [36] first proposed the idea of the conventional neural network in the late 1990s. The proposed CNN and today's Alex Net have almost the same architecture. But it did not gain much popularity because of lack of data availability for training of neural network in solving vision problems. Obviously with passage of time and through the internet explorations, a huge amount of data exchange started especially for the neural networks, which helped computer scientists and researchers to train their neural networks on large amount of data. The gathered data was well organized and available in the form of manageable databases. Because of the availability of large amount of data for training, the performance of neural networks increased and ultimately it reduced the error rate by a noticeable amount. Despite of all the revolutions in technology mainly in computer vision and CNN we are still far away from making a vision system to be intelligent and responsible enough as a human being in order to understand a scene and recognition & detection of particular targets in images.

Due to recent advances and innovations in the field of Computer Vision and for the high demands of Earth Vision applications a lot studies and work have been done in order to recognize & detect targets in aerial images. But most of these studies attempt to transfer the knowledge and techniques especially the algorithms used for detection of targets in natural scene/imagery to the aerial image domain. Most of the Earth Vision researchers have used the approaches based on fine-tuning the networks that are already pre-trained on large-scale image datasets (e.g., ImageNet and MSCOCO) for detection of required targets in the aerial domain because of the successes of deep learning-based algorithms for target detection [40]. So such approaches based on pre training and fine-tuning are a reasonable avenue to explore especially in the case of Satellite/Aerial imagery.

Most of the studies particularly based on image segmentation and classification reveals that the method used for detection and classification of targets in satellite/ aerial

imagery is much distinguished from the conventional methods of target recognition & detection in natural imagery in the following respects:

In satellite/aerial imagery the scale variations of object instances are huge as compared to natural imagery. The main reason behind it is the variation in both the spatial resolutions of sensors and also in the size inside the same target category.

In aerial images there are a lot of target instances especially of small size target instances, like, the ships and the vehicles in satellite images. Moreover, target instances in aerial images have unbalanced frequencies , for example, it is possible that some small-size e.g. 800X800 images contain thousand plus instances, while some large-size images like size of 3000X3000 may contain only a hand full of small instances. This is more likely unbalances target instances distribution in satellite/aerial imagery [40].

The orientations of Targets in satellite/aerial images are often arbitrary and there are some examples in which target instances have extremely large aspect ratio, such as a bridges and roads etc.

3.1 Brief Description of Datasets

3.1.1 DOTA [40]

To advance the research for recognize and detect targets in aerial images, Earth Vision introduces a large-scale “Dataset for Object detection in Aerial images (DOTA)” [40] constructed by Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie,Jiebo Luo, Mihai Datcu, Marcello Pelillo, Liangpei Zhang. This dataset contains about 2806 aerial images are collected from different sensors and regions with crowdsourcing. Each image contains targets of different scales, orientations and shapes is of the size about 4000 X 8000 pixels. The annotation is done by experts in aerial image understanding and interpretation, according to 15 common object categories. This annotated dataset has total of 188,282 instances, each of which is labeled by an oriented bounding box.

Figure 3.1: Sample Images of DOTA Dataset

3.1.2 NWPU VHR-10 [41]

NWPU VHR_10 is another dataset data and contain very-high-resolution (VHR) remote sensing images dataset was constructed by Dr. Gong Cheng et al. from Northwestern Poly technical University (NWPU).This is basically a ten categorical geospatial object detection dataset used for research purposes only. Actually this dataset is subset of a dataset that has 45 categories. In this dataset there are totally 800 VHR remote sensing images, and contain two main folders first one is "negative image set" includes 150 images that do not contain any targets of the given classes and the second one is "positive image set" includes 650 images with each image containing at least one target to be detected. These images were cropped from Google Earth and Vaihingen data set and then manually annotated by experts. The folder "ground truth" contains 650 separate text files and each one corresponds to an image in "positive image set" folder. Each line of those text files defines a ground truth bounding .The 650 images contain instances of each category like 302 instances of ships, 477 vehicles instances, 124 instances of bridges, 224 harbors instances, 163 ground track fields instances, 390 instances of baseball diamonds, 524 tennis courts instances, 159 instances of basketball courts, 655 storage tanks instances, and finally 757 instances of airplanes.

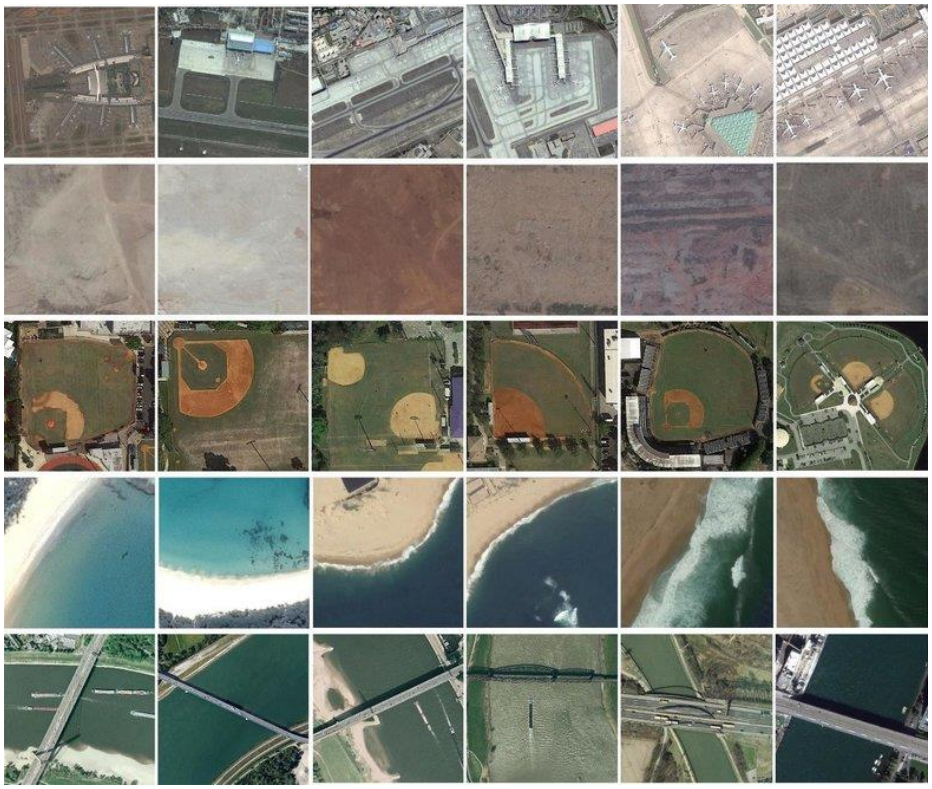

Figure 3.2: Sample Images of NWPU_10 Dataset

Despite of these distinct features and difficulties of satellite and aerial images a lot of efforts have been put to make target recognition and detection in aerial imagery easier. So there are a lot of datasets of satellite/aerial images are now available for recognition and detection of particular targets in Earth Vision with the good degree of generalizability across datasets. In order to accomplish the task of automatic target recognition and detection in aerial images we consider two datasets and their comparison is shown in following table.

Table 3.1: Comparison of Datasets

S.NO	Data Set	Object Category	Image Size	Annotation	Aim of Dataset
1	DOTA: A Large-scale Dataset for	1:Plane 2:Ship 3:Storage tank	4000 X 4000 To 8000	Annotations are given in the form of a text file as shown below: **'image source': image source'gsd':gsd**	classification task and scene recognition task

	Object Detection in Aerial Images	4:Baseball diamond 5:Tennis court 6:Basketball court 7:Ground track field 8:Harbor 9:Bridge 10:Large vehicle 11:Small vehicle 12:Helicopter 13:Roundabout 14:Soccerball field 15:Swimming pool	X 8000	**x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4,** category, difficult **x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4,** category, difficult ... where (x_i, y_i) denotes the positions of the oriented bounding boxes' vertices in the image, and meanwhile a difficult label is provided which indicates whether the instance is difficult to be detected(1 for difficult, 0 for not difficult)	
2	NWPU VHR-10 dataset	1.ship 2. vehicle 3.bridge 4.harbor 5.ground track field	from 533×597 to 1728×1028	text files defines a ground truth bounding box like: (x1,y1) ,(x2,y2),a where (x1,y1) denotes the top-left coordinate of the bounding box,	The main goal of this dataset is research in remote sensing image scene

		6. baseball diamond s 7.tennis court 8.basketball court 9.storage tank 10.airplane		(x2,y2) denotes the right-bottom coordinate of the bounding box, and a is the object class.	understanding and classification.

3.1.3 DOTA VS NWPU

In the following Figure, we compare the categories of DOTA with NWPU VHR-10, which has the largest number of categories in previous aerial object detection datasets. Note that DOTA surpass NWPU VHR-10 not only in category numbers, but also the number of instances per category.

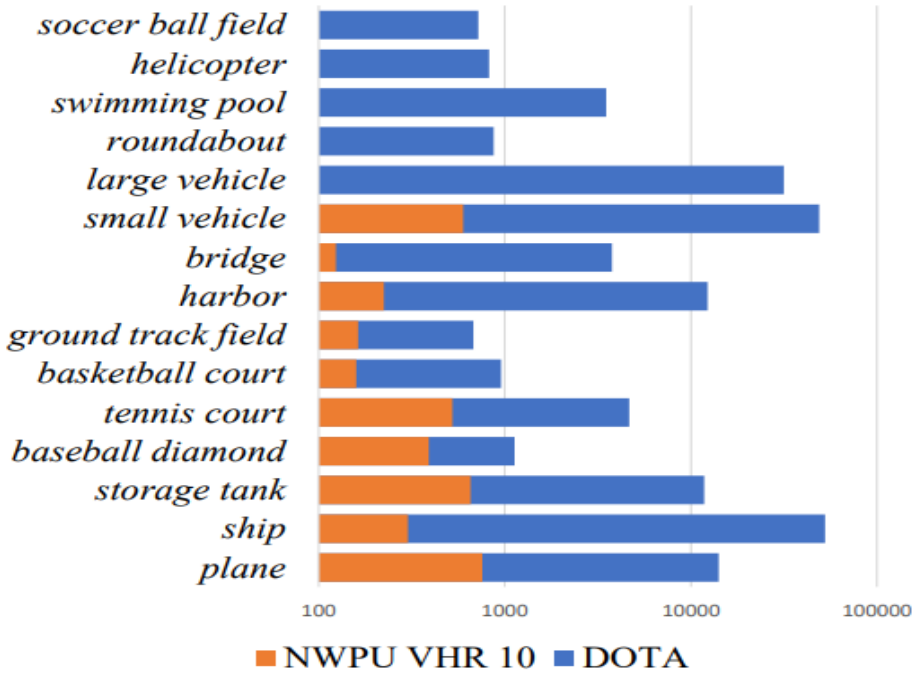

Figure 3.3: DOTA VS NWPU [40]

Table 3.2: List of papers based on these datasets:

S. No.	Author	Year	Technique	Data set	Reported results map

1	Gui-Song Xia1 , Xiang Bai	May 21, 2019	YOLOv2 Faster R-CNN	Dota (A Large-scale Dataset for Object Detection in Aerial Images)	39.2 % 60.46 %
2	Li Zhuo	January 18, 2018	CNN (Google Net Fine-tuning)	Vehicle Dataset (13,700 Aerial images)	98.6%
3	Xiaobing Han ,Yanfei Zhong	June 28,2017	Faster R-CNN	NWPU(North Western Polytechnic University) Dataset	85%

This is the list of papers based on the two datasets and different CNN algorithm results. Based on these papers we evaluated our proposed framework on DOTA dataset, which contains 2806 aerial images with pre-divided 1411 training images, 458 validation images and 937 testing images. Those DOTA images are obtained from different sensors and platforms with crowdsourcing and the size ranges from 800 _ 800 to 4000 _ 4000 pixels. DOTA consists of 15 common categories, namely, plane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge, large vehicle, small vehicle, helicopter, roundabout, soccer ball field and swimming pool. The fully annotated DOTA dataset contains 188,282 instances, each of which is labeled by an oriented quadrilateral instead of an axis-aligned one, which is typically used for object annotation in natural scene images. Another common geospatial object detection dataset is NWPU VHR-10 [44], which contains 800 images in 10 categories with a total of 3651 instances. The average size of NWPU VHR-10 is 1000 _ 1000 pixels. Compared

with NWPU, DOTA is a larger annotated dataset for multi-class geospatial object detection, which has more complex backgrounds, larger image size and denser object distribution thus more reflective of the real-world applications. Therefore, the evaluation on DOTA can better verify the effectiveness and robustness of our proposed network.

Chapter 4: PROPOSED METHODOLOGY

This chapter includes the proposed architecture and all the baseline methods which are used for detection in aerial images especially DOTA dataset. Our proposed methodology consists of three main modules. The first module is a dataset acquisition module in which we change the given dataset according to our model respectively. Second module is basically training of the model. From this module a final trained model is obtained which is then used to test the images in the next module. This module takes the input image and detect the objects with in it and classify the detected objects as one of the 15 categories. These three models are explained in detail below. Figure 4.1 shows the flow diagram of our proposed technique which clearly shows all the three blocks of the model and how model is trained and then used for classify and localize the objects. The sections below discuss the three modules of the proposed technique in detail.

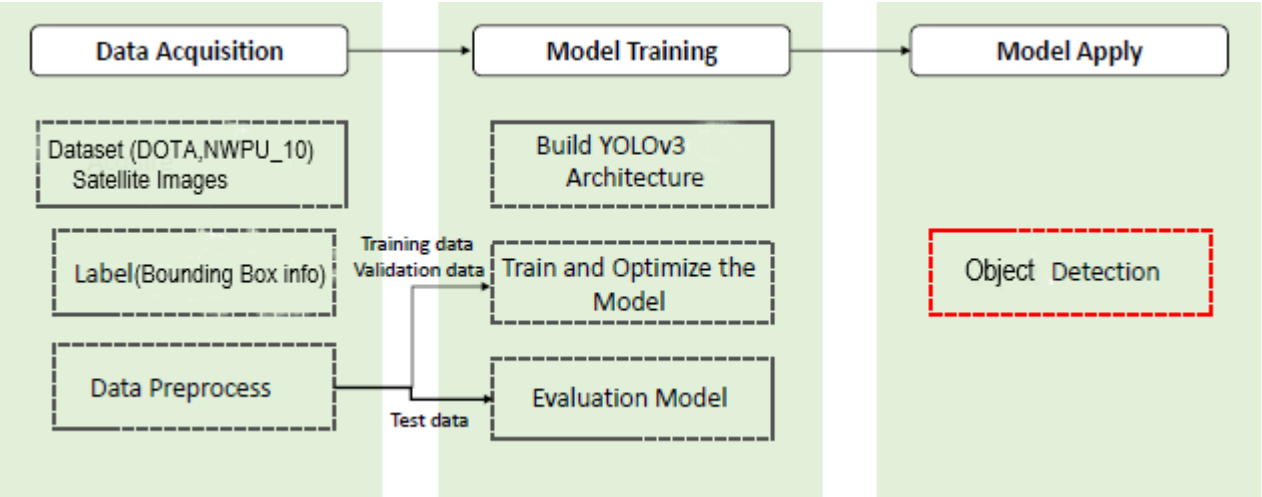

Figure 4.1: Flow Chart of proposed technique

4.1 Object Detection

Beside simple image classification, there's no shortage of fascinating problems in computer vision, with object detection being one of the most interesting. Most commonly it's associated with self-driving cars where systems blend computer vision, LIDAR and other technologies to generate a multidimensional representation of road with all its

participants. On the other hand object detection is used in video surveillance, especially in crowd monitoring to prevent terrorist attacks, count people for general statistics or analyze customer experience with walking paths within shopping centers.

In this task we've got an image and we want to assign it to one of many different categories (e.g. car, plane, ship, bridge), so basically we want to answer the question "What is in this picture?" Note that one image has only one category assigned to it. After completing this task we do something more difficult and try to locate our object in the image, so our question changes to "What is it and where it is?" This task is called object localization. So far so good, but in a real-life scenario, we won't be interested in locating only one object but rather multiple objects in one image [43]. For example let's think of a self-driving car, that in the real-time video stream has to find the location of other cars, traffic lights, signs, humans and then having this information take appropriate action. It's a great example of object detection. In object detection tasks we are interested in finding all object in the image and drawing so-called bounding boxes around them. There are also some situations where we want to find exact boundaries of our objects in the process called instance segmentation, but this is a topic for another post.

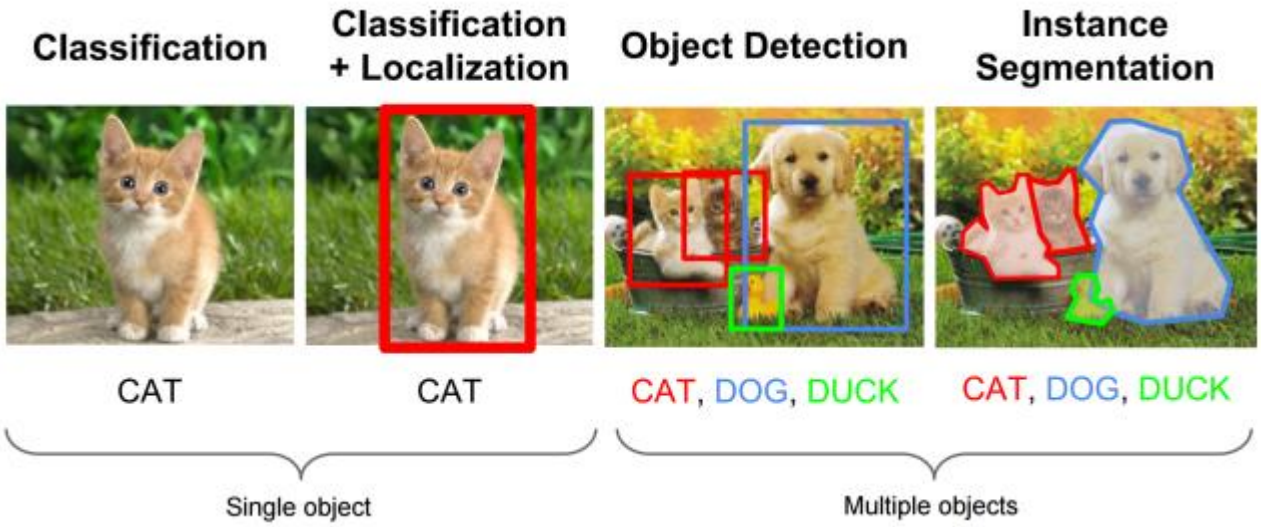

Figure 4.2: Object Detection [43]

4.2 YOLO Algorithm

There are a few different algorithms for object detection and they can be split into two groups:

Algorithms based on classification – they work in two stages. In the first step, we're selecting from the image interesting regions. Then we're classifying those regions using convolutional neural networks. This solution could be very slow because we have to run prediction for every selected region. Most known example of this type of algorithms is the Region-based convolutional neural network (RCNN) and their cousins Fast-RCNN and Faster-RCNN [44].

Algorithms based on regression – instead of selecting interesting parts of an image, we're predicting classes and bounding boxes for the whole image **in one run of the algorithm**. Most known example of this type of algorithms is **YOLO (You only look once)** commonly used for real-time object detection.

Before we go into YOLOs details we have to know what we are going to predict. Our task is to predict a class of an object and the bounding box specifying object location. Each bounding box can be described using four descriptors:

- Center of a bounding box ($\mathbf{b_x}$ $\mathbf{b_y}$)
- Width ($\mathbf{b_w}$)
- Height ($\mathbf{b_h}$)
- Value **c** is corresponding to a class of an object (i.e. vehicle, plane etc…).

We've got also one more predicted value p_c which is a probability that there is an object in the bounding box, we need this because.

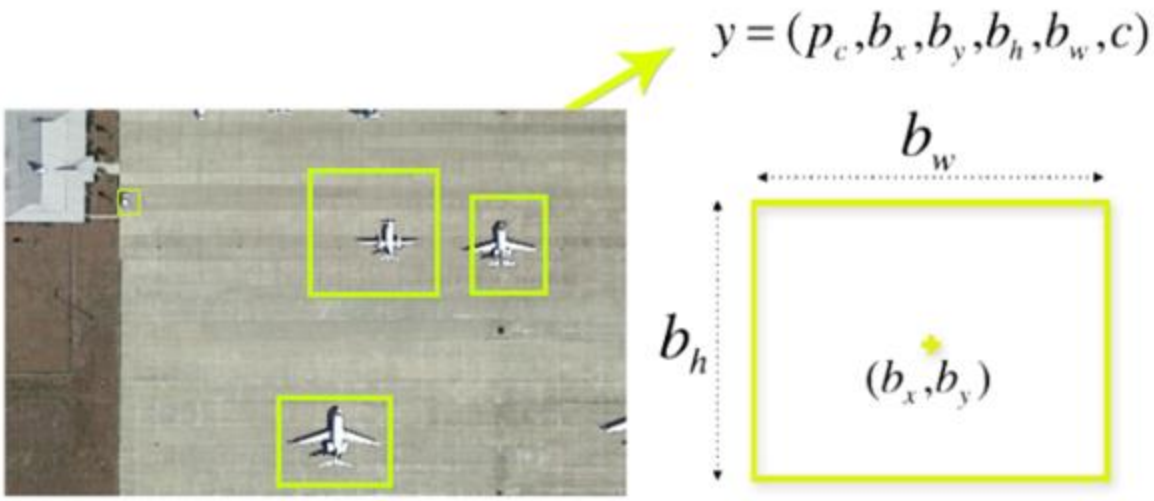

Figure 4.3: YOLO bounding box

Like mentioned before with YOLO algorithm we're not searching for interested regions on our image that could contain some object. Instead of that we are splitting our image into cells, typically its 19×19 grid. Each cell will be responsible for predicting 5 bounding boxes (in case there's more than one object in this cell). This will give us 1805 bounding boxes for an image and that's a really big number!

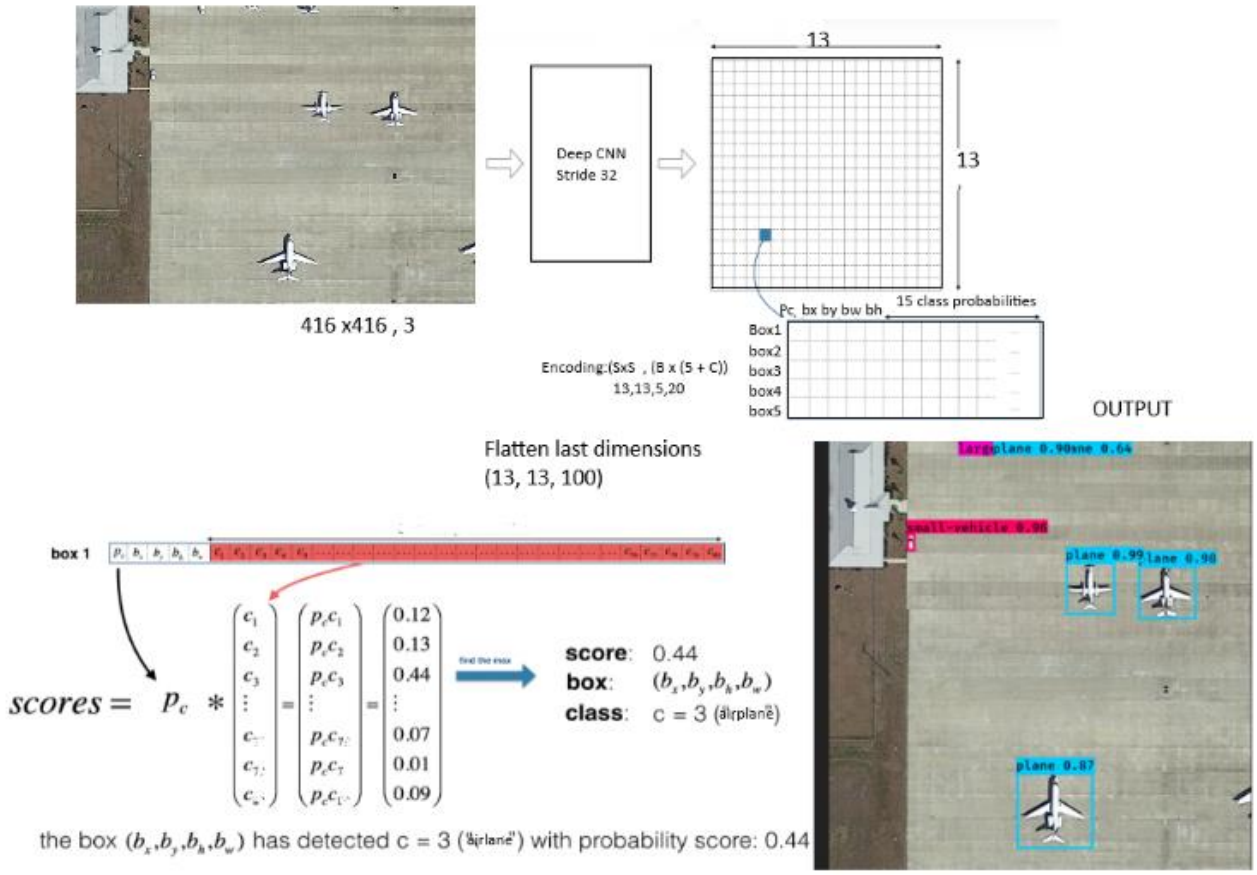

Figure 4.4: Yolo Algorithm

Majority of those cells and boxes won't have an object inside and this is the reason why we need to predict p_c. In the next step, we're removing boxes with low object probability and bounding boxes with the highest shared area in the process called **non-max suppression**.

Figure 4.5: YOLO Non-Max Suppression

4.3 Different Version of YOLO

4.3.1 YOLO v1 Architecture

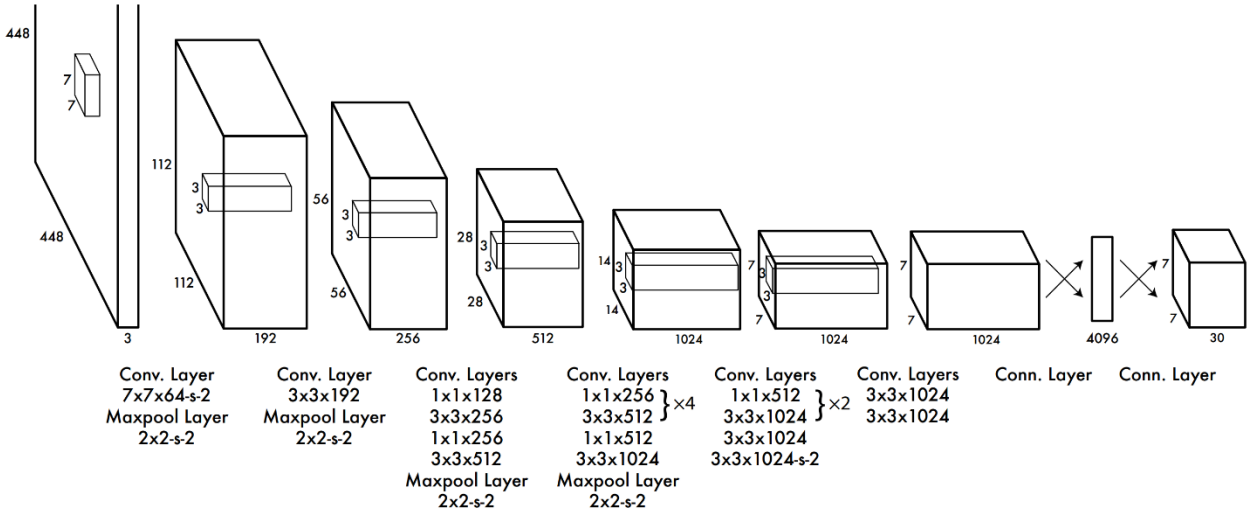

Figure 4.6: Yolo v1 architecture [44]

It uses the Darknet framework trained on the dataset of ImageNet-1000. This works as mentioned above, but because of this, the use of the YOL v1 is limited. Small objects could not be found if they appeared as a cluster. This architecture found it

difficult to generalize objects when the image is different from the trained image in other dimensions. The main issue is to locate objects in the image of the input [8].

***Problem with YOLO* v1**

Figure 4.7: The picture above shows that YOLO Version1 is limited by object proximity. YOLO only senses five Santa's, but 9 Santa's from the lower left corner [41]

4.3.2 YOLO v2

At the end of 2016, Joseph Redmon and Ali Farhadi released the second YOLO release, called YOLO9000. More quickly and gradually, the new FASTER R-CNN update involves an object sensing algorithm that uses a Regional Proposal Network to identify the image input objects [45] and SSD (single shot multi box detector).

YOLO updates v2 to the YOLO standard batch: normalizes the output layer with the activations changed and slightly scaled. Batch standard reduces unit costs for the hidden layer and improves the stability of the neural network. Adding the batch of standardization.

YOLO updates to the YOLO v2:

- **Batch Normalization**: It normalizes input layer by slightly modifying and scaling activations. Unit value shifts for the hidden layer were avoided by batch standardization and neural network consistency thus improved. Increased MAP (mean-avg-precision) with 2 percent batch-standardization in architecture. It has also helped to regularize and update the program.
- **Higher Resolution Classifier:** Yolov2 increased input size from 224 * 224 in YOLO v2 to 448 * 448. The rise in image output has improved up to 4% on the MAP (mean average accuracy). The rise in output size is used during the ImageNet database practice of the YOLO v2 architecture DarkNet 19 .
- **Anchor Boxes:** The introduction of the anchor boxes is one of the most notable changes that may be seen in YOLO v2. In one single framework, YOLO v2 classifies and predicts. The anchor boxes are designed with clustering (k-means clustering) to predict bounding boxes and this anchor boxes is designed for a specific set of data [44].
- **Fine-Grained Features:** The identification of smaller objects on the image is one of the key issues to be addressed in the yolo v1. This is decided in the YOLO v2 and the image is divided into 13* 13 grid cells, which is less than the previous version. It helps the yolo v2 to recognize or locate the smaller objects in the picture as well as the larger ones.
- **Multi-Scale Training:** In YOLO v1 there are limitations in detecting objects of different input sizes that suggest that YOOLO has trouble detecting the same object in a larger image when equipped with small images of a certain object. This is largely solved in YOLO v2 and is trained with random pictures ranging from 320* 320 to 608* 608[5] in different dimensions. It allows the network to reliably learn and predict the images from various input dimensions.
- **Darknet 19:** YOLO v2 uses Darknet19 architecture for classifying objects with 19 convolutional layers and 5 max pooling layers. The Darknet 19's architecture was displayed below. Darknet is a Clanguage and CUDA neural network system. Very quickly it's very important in object detection for real time prediction.

Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

Figure 4.8: Dark net 19 architecture [45]

Progresses are better, faster and stronger in several categories in YOLO v2 as stated in[45]. The network now identifies and classifies objects of various configurations and measurements with Multi-Scale Learning. The identification of smaller objects with far better accuracy than its predecessor version was significantly improved by YOLO v2.

4.3.3 YOLO v3

The previous version has been improved for an incremental improvement which is now called YOLO v3. As many object detection algorithms are been there for a while now the competition is all about how accurate and quickly objects are detected. YOLO v3 has all we need for object detection in real-time with accurately and classifying the objects. The authors named this as an incremental improvement [44].

Here we will have look what are the so called Incremental improvements in YOLO v3.

The Algorithm

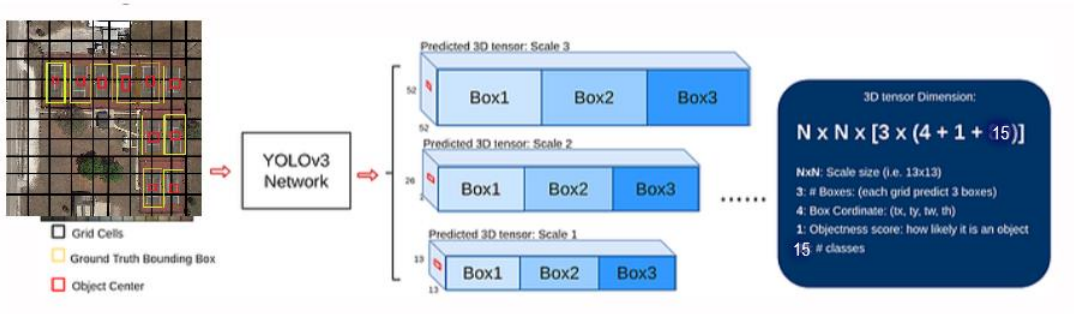

Figure 4.9: YOLOv3 Algorithm

- **Darknet-53:** the predecessor YOLO v2 used Darknet-19 as feature extractor and YOLO v3 uses the Darknet-53 network for feature extractor which has 53 convolutional layers. It is much deeper than the YOL v2 and also had shortcut connections. [6]. Darknet-53 composes of the mainly with 3x3 and 1x1 filters with shortcut connections.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 4.10: Darknet-53 [46]

Object detection decreases the human resources in many areas. Object detection in real-time and reliably is one of the main requirements in the world where self-driving cars are becoming a reality. The object detection algorithms like YOLO v3, faster R-CNN, SSD and many have a wide variety of improvements. Minimal improvements can change the whole perception of these algorithms within the real world.

First, during training, YOLOv3 network is fed with input images to predict 3D tensors (which is the last feature map) corresponding to 3 scales, as shown in the middle one in the above diagram. The three scales are designed for detecting objects with various sizes. Here we take the scale 13x13 as an example. For this scale, the input image is divided into 13x13 grid cells, each grid cell corresponds to a 1x1x255 voxel inside a 3D tensor. Here, 255 comes from (3x (4+1+15)). Values in a 3D tensor such as bounding box coordinate, abjectness score and class confidence are shown on the right of the diagram.

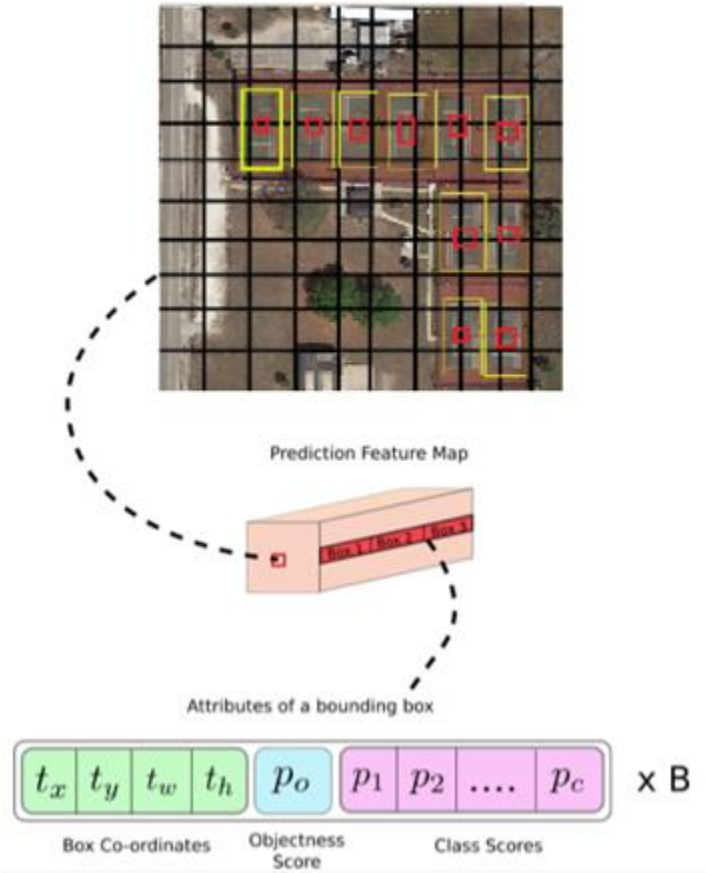

Figure 4.11: YOLO V3 BBOX

Second, if the center of the objects ground truth bounding box falls in a certain grid cell (i.e. the red one on the tennis court image), this grid cell is responsible for predicting the object's bounding box. The corresponding abjectness score is "1" for this grid cell and "0" for others. For each grid cell, it is assigned with 3 prior boxes of different sizes. What it learns during training is to choose the right box and calculate precise offset/coordinate. But how does the grid cell know which box to choose? There is a rule that it only chooses the box that overlaps ground truth bounding box most.

Lastly, how to choose the initial size of those 3 prior boxes? We used K-mean clustering to classify the total bounding boxes from COCO dataset to 9 clusters before training. This results in 9 sizes chosen from 9 cluster, 3 for 3 scales. This prior information is helpful for the network to learn to compute box offset/coordinate precisely because intuitively, bad choice of box size make it and longer for the network to learn.

- **The Network Architecture**: YOLOv3's network architecture is a feature-learning based network that adopts 75 convolutional layers as its most powerful tool. No fully-connected layer is used. This structure makes it possible to deal with images with any sizes. Also, no pooling layers are used. Instead, a convolutional layer with stride 2 is used to down sample the feature map, passing size-invariant feature forwardly. In addition, a ResNet-alike structure and FPN-alike structure is also a key to its accuracy improvement.
- **Scales: handling objects of different sizes:** There are objects of different sizes on the images. Some are big and some are small. It is desirable for the network to detect all of them. Therefore, the network needs to be capable of "seeing" objects that are of different sizes. As the network goes deeper, its feature map gets smaller. That is to say, the deeper it goes, the harder it is to detect smaller objects. Intuitively, it is better to detect the objects at different feature maps before small objects end up disappearing. As in SSD, object detection is carried out on different features maps in order catch various scales.

However, the features are not absolutely relevant at different depth. What does this mean? Well, with network depth increasing, the features change from low-level features

(edges, colors, rough 2D positions. etc) to high-level features (semantic-meaningful information: dog, cat, car. etc) with depth increasing. So making predictions on feature maps at different depth does sound it is able to do detection for multi-scale objects, but actually it is not as accurate as expected.

In YOLOv3, this is improved by adopting an FPN-like structure.

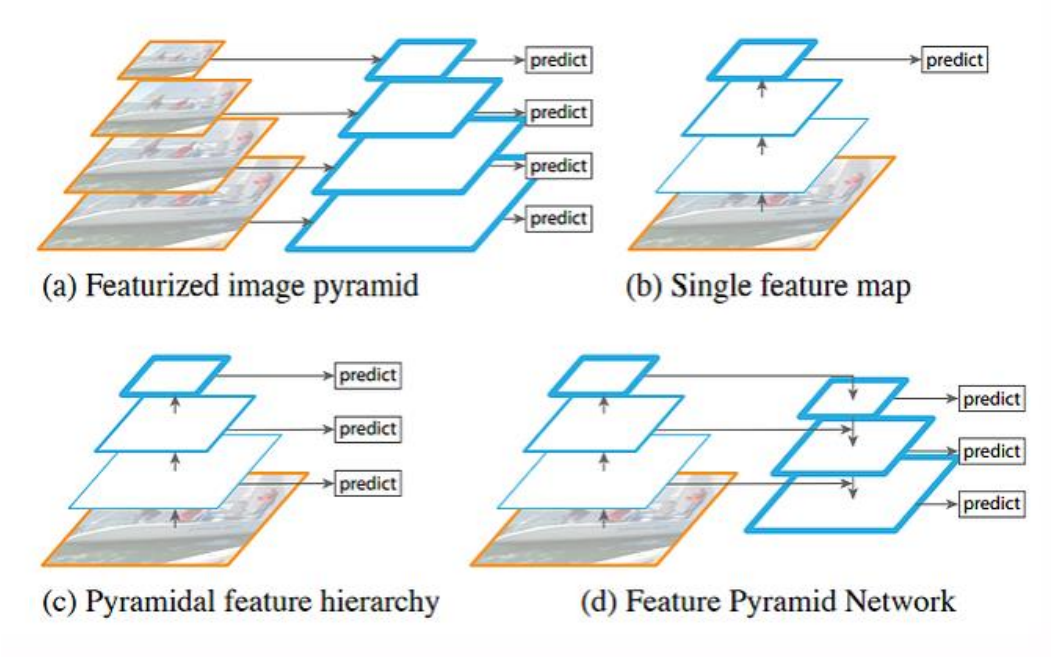

Figure 4.12: Multi-scale Feature Learning Illustration [46]

As shown on the above illustration, there are 4 basic structure for multi-scale feature learning.

The most straightforward one. Construct an image pyramid and input each pyramid level to individual network specially designed for its scale. As a result, it is slow because each level needs its own network or process.

The prediction is done at the end of the feature map. This structure cannot handle multiple scales.

The prediction is done on feature maps at different depth. This is adopted by SSD. The prediction is done by using the features learned so far and further features at deeper layers cannot be utilized.

Similar to (c) but further features are utilized by up sampling the feature map and merged with current feature map. This is fascinating because it let current feature map

to see its features in "future" layers and utilize both to do accurate prediction. With this technique, the network is more capable to capture the object's information, both low-level and high-level.

In YOLOv3, there are 3 scales used in (d) form. This helps detect small objects effectively. As shown in the figure below, small cars and people can be detected successfully.

- **Prediction across different scales**
- YOLO v3 makes 3 different scales prediction. The detection layer is used to detect feature maps with stages **32, 16, 8** and three different sizes. We therefore detect scales **13 x 13, 26 x 26** and **52 x 52** with an input of **416 x 416**.
- The network samples the image input to the first detector layer when a detection is performed using layer charts with phase 32. In addition, layers are sampled with a factor of 2 and are combined with the characteristics of previous layers with identical map sizes. Another layer detection with step 16 is now made. This performs the same up sampling process and makes the final detection on stage 8.
- Each cell forecasts 3 border boxes on 3 anchors at each level, making the total number of anchors used 9. (For different scales the anchors are identical)

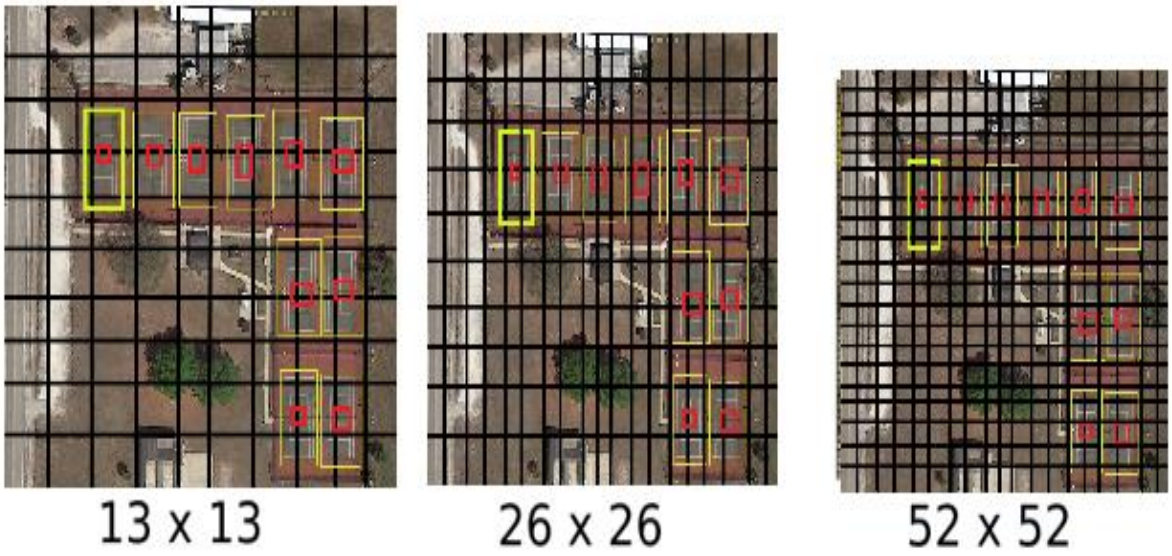

Figure 4.13: Prediction across Different Scales

This helps to detect small objects, a common complaint about previous YOLO versions. The sampling of smaller objects can help the network learn from fine grain features.

- **ResNet-alike structure: a better way to grasp good features**

In YOLOv3, a ResNet-alike structure (called Residual Blocks in the YOLOv3 Architecture Diagram) is used for feature learning. Basically a Residual Block consists of several convolutional layers and shortcut paths. An example of a shortcut path is illustrated below.

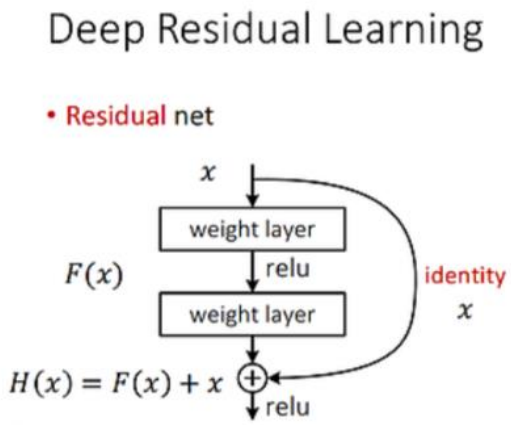

Figure 4.14: Deep Residual Learning [46]

The curved arrow on the right represents a shortcut path. Without this, it is a classic CNN network, which learns the feature one by another. As the network goes deeper, the harder it is to learn the features well. If we add a shortcut as shown above, the layers inside the shortcut learns what to add to the old feature in order to generate better feature. As a result, a complex feature H(x), which used to be generated standalone, is now modeled as H(x)=F(x)+x, where x is the old feature coming from the shortcut and F(x) is the "supplement" or the "residual" to learn now. This makes it easier for the network to learn the features stably, especially in very deep networks. This technique changes the goal of learning. Thus, instead of learning a complete complex feature, the new goal is to learn the supplemental "residual" that is used to be added up to the old feature, which simplified complexity for learning.

- **No softmax layer: multi-label classification**

Softmax layer is replaced by 1x1 convolutional layer with logistic function. By using a softmax, we have to assume that each output only belong to exactly ONE of the

classes. But in some dataset or cases where the labels are semantically similar (i.e. Woman and Person), training with softmax might not let the network generalize the data distribution well. Instead, a logistic function is used to cope with multi-label classification.

4.3.4 Comparison between YOLO V1, V2 and V3

Following table gives a quick overview of above theory that includes the architecture details, working strategies, drawbacks and improvements from version to version in YOLO.

Table 4.1: Comparison between all versions of YOLO

YOLO V1	YOLO V2	YOLO V3
The first version of yolo consists of total 26 layers	Compared with YOLO v1 it has 30 layers. No fully connected layer available	Darknet 53. Neural network of 106 layers
24 conv Layers followed by two layers with full connection	Batch Normalization layers after each Conv layer are included	3scalar detection to detect small to large objects
	Implementation of anchor boxes. Anchor boxes are predefined Darknet boxes that give the network a sense of the relative position and dimensions Of the objects to be identified. It must be measured with the train set.	9 boxes taken from the anchor; 3 by size. Further border boxes than YOLO9000 & YOLOv1 Are therefore planned.

	For training images from 320 to 608 random dimensions are taken	Problem multi-class turned into multi-label
	Several labels can, but still a multiclass, Be provided for the same objects.	Some of the error function changes
The main problem with YOLOv1 is that it is not capable of recognizing very small objects	Even bad for small objects	Really great with small items

Based on above comparison we proposed YOLOv3 network for our research. YOLOv3 is the third object detection algorithm in YOLO (You Only Look Once) family. It improved the accuracy with many tricks and is more capable of detecting small objects. Chapter 5, gives a detailed overview of our experiments and results in which firstly, the remote sensing image data set of complex background, multi-scale target, multi-objective, multi-category and different perspectives are constructed independently, which lays a foundation for the training of the model. Then the YOLOv3 algorithm is improved for the target characteristics in the data set, so that the model can extract more deep-separated features of the target and play a better training effect. Finally, the effectiveness and significance of the algorithm are verified by comparison with other algorithms.

Chapter 5: EXPERIMENTS AND RESULTS

In this chapter, we evaluate the experiment done and its results on DOTA and NWPU_10 data set. A brief overview of both the datasets is given and then evaluation parameters are discussed. Results of localization and classification are shown in the form of some curves. The results of are also depicted in the visual image form.

5.1 Dataset

- **DOTA: A large scale dataset for detection in Satellite Images**

DOTA data is acquired by the Google Earth, some are taken by satellite JL-1, and the others are taken by satellite GF-2 of the China Centre for Resources Satellite Data and Application. Different images are taken in different lighting condition to observe the effect of light on the scene and on the object detection. Recently an updated version of DOTA that is DOTA-v1.5 is released and we are using the updated version of DOTA in our experiment. Figure 5.1 shows randomly chosen images from DOTA-v1.5 data set and it ground truth bounding box representation. DOTA-v1.5 consists of total 2806 aerial image that are fully annotated with 16 categories and contain almost 188, 282 instances. The size of every image is 4000x4000. The dataset is divided into 1411 training images, 560 images and rest of almost 900 images for testing [41].

(a) (b) (c)

Figure 5.1: DOTA-v1.5: A large scale dataset for object detection in Satellite Images [41]

NWPU_10 VHR dataset

Apart from DOTA-v1.5 data, we used another satellite images dataset which consisted of total 800 very-high-resolution satellite images. From 800 images, 650 are positive image set that contain 10 same categories as that of DOTA-v1.5 and rest 150 are negative image set. Figure 5.2 shows some random images from the nwpu_10 dataset.

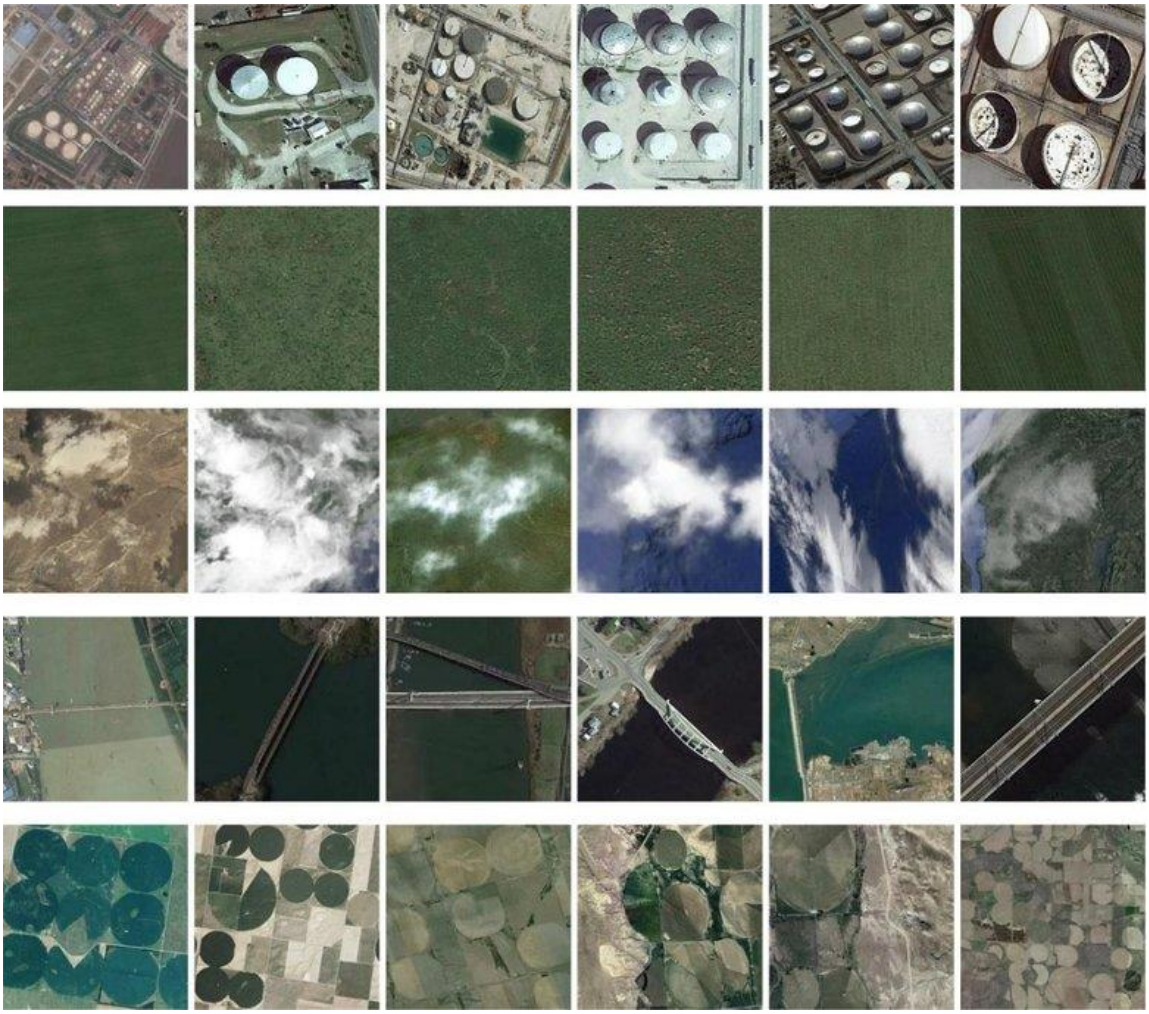

Figure 5.2: Sample Images from NWPU_10 Dataset [42]

5.2 Localization and classification

The basic purpose of detection is to localize and classify all the instances of 16 categories from input images. Darknet-53 with additional 53 layers is used as base network to extract features maps. The reason behind choosing Dark net over other CNN

architectures was that it is most speedy CNN architecture with comparative accuracy. There are two types of blocks in deeper models, one is plain block and the other is residual block. Plain blocks maps input, X directly to the H (x) by passing through some stacked layers. Residual blocks use residual mapping of F(x) = H(x) – x, instead of direct mapping to H(x) we try and learn some function F(x) that needs to be added in x to get H(x). In this way we can find H (x) indirectly. A skip connection is built from x to be added to next block having the residual content including weight layers. If no weight layer is used in the network, then skip connection acts as an identity, but when weights are included, then learned weights are added in the next block. Darknet-53 is composed of residual blocks stacked on top of each other.

To measure the efficiency of our experiment we are using three loss parameters. These parameters are used to calculate three different types of losses during the training of algorithm. The loss parameters used in this research include: The localization loss (errors between the predicted boundary box and the ground truth), the confidence loss (the object ness of the box) and the classification loss. The first one penalizes the object ness score prediction for bounding boxes responsible for predicting objects (the scores for these should ideally be 1).The second one for bounding boxes having no objects, (the scores should ideally be zero). The last one penalizes the class prediction for the bounding box which predicts the objects.

5.3 Detection Results

Proposed YOLOV3 architecture uses a variant of Dark net, which originally has 53 layer network trained on ImageNet. For the task of detection, 53 more layers are stacked onto it, giving us a 106 layer fully convolutional underlying architecture for YOLO. Base network extract features from their respective input images. The backbone of yolo v3 introduces a residual structure, so now the network can get very deep. Inside the entire v3 structure, there is no pooling layer and full connectivity layer. In the forward propagation process, the size transformation of the tensor is realized by changing the step size of the convolution kernel, such as stride= (2, 2), which is equivalent to reducing the edge length of the image by half (i.e., the area is reduced to the original 1/4). Yolo_v3 is also the same as v2, and the backbone will shrink the output feature

map to 1/32 of the input. Therefore, it is usually required that the input picture be a multiple of 32.

Yolo v3 outputs 3 feature maps of different scales, such as y1, y2, and y3 shown in the figure above. This is also one of the few improvement points mentioned in the v3: predictions across scales

Draws on FPN (feature pyramid networks) and uses multiple scales to detect targets of different sizes. The finer the grid cells, the more accurate the detection. Fine objects. The depth of all three scales 1,2 and 3 are all 60 because yolo v3 is set to predict 3 boxes per grid cell, so each box needs to have (x, y, w, h, confidence) five basic parameters, and then has a probability of 15 categories. So 3 * (5 + 15) = 60. This is the 60 and the rule of side length is 13:26:52.

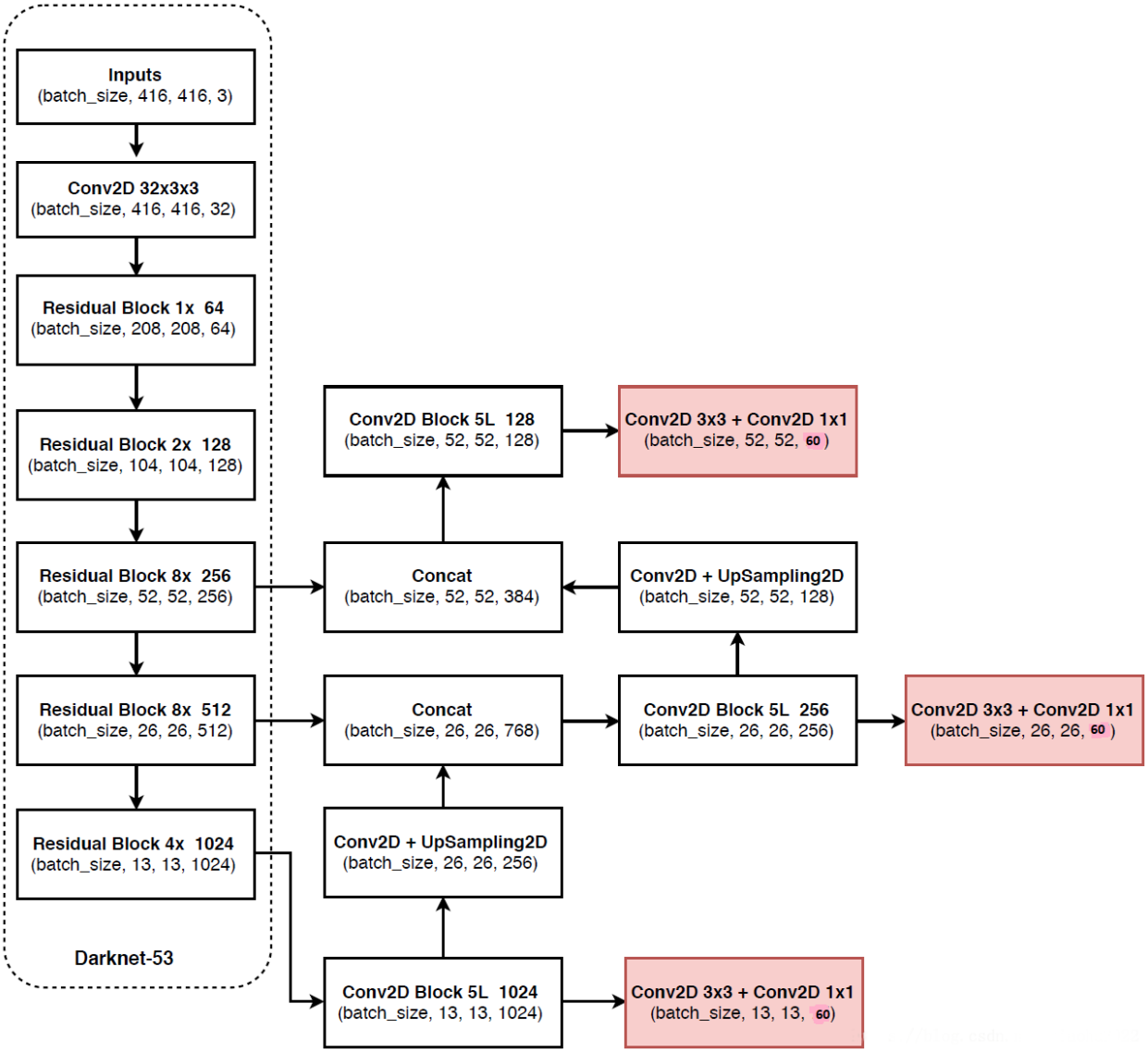

Figure 5.5: YOLO V3 architecture using DarknetNet-53

5.4 Evaluations

5.4.1 Evaluation metrics

There are two types of evaluation metrics we chose for result evaluation. First is training and validation loss plots of model for comparison of results. Second is AP (Average precision) is a popular metric in measuring the accuracy of object detectors like Faster R-CNN, SSD, etc. Average precision computes the average precision value for recall value over 0 to 1. It sounds complicated but actually pretty simple as we illustrate

it with an example. But before that, we will do a quick recap on precision, recall, and IOU first.

- **Loss:** Basically the loss function for any model is one of the important criterion for evaluating the performance of that model during training and testing. In YOLO, [18] the loss function is sum of three types of mean square errors that are object ness score, second one is bounding box predictions and the last one is classification error so the whole loss function based on these errors can be defined by following mathematical equation:

$$YOLO\ \ Loss = \quad \text{Error}coordinate\ \ + ErrorIOU + Errorclass \qquad (1)$$

Where in first part of above equation Errorcoord the value of the four of bounding box error, S2 is the number of grid cells in which any input image is divided by the network, and B basically represents the of bounding boxes generated from each grid cell by network. Referring to our approach we change the parameters in the original YOLOV3 model according to our dataset, S = 13, and B = 9 .and in the second part of first equation if there is an object in grid cell then it is denoted by $\mathbf{1}ij$ = 1 *which means* that the object is present in the jth bounding box of grid cell ith, otherwise $\mathbf{1}ij$ = 0 which means there is no object in that bounding box. (xi^, yi ,^ wi^, hi)^ are values of the four coordinates that are height, and width and central coordinates of the predicted bounding box. And (xi, yi, wi, hi) are four coordinate Values of actual bounding box respectively.

Where the parameter error IOU *represents* the value of the $\lambda noobj$. For our implementation we choose a threshold value = 0.5 for IOU. Whereas *Ci* represents the class confidence score for both predicted and actual bounding boxes.

The third and the last part of equation is used to measure the class error in which c denotes from which specific class does the detected target belongs to. *pi* (*c*) represents the probability of both actual and predicted probability that the object belonging to this specific class c is in grid cell i. The *class error* for grid i is the sum of classification errors for all the objects in the grid cells.

- **Precision** measures how accurate is your predictions. I.e. the percentage of your predictions are correct.

$$\text{Preccision} = \frac{Tp}{\text{Tp} + \text{Fp}} \qquad (2)$$

- **Recall** measures how good you find all the positives. For example, we can find 80% of the possible positive cases in our top K predictions.

$$Recall = \frac{Tp}{\text{Tp} + F_N} \qquad (3)$$

- **IOU (Intersection over union):**IOU measures the overlap between 2 boundaries. We use that to measure how much our predicted boundary overlaps with the ground truth (the real object boundary). In some datasets, we predefine an IOU threshold (say 0.5) in classifying whether the prediction is a true positive or a false positive.

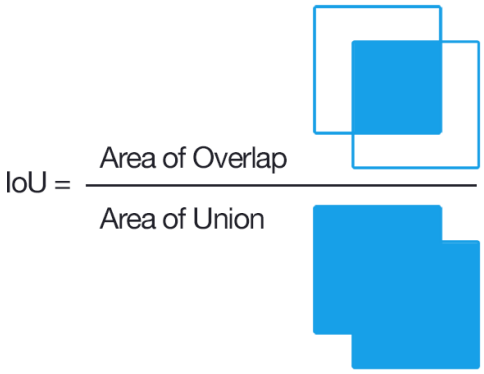

Figure 5.3: IOU Presentation [47]

We have "a match" when they share the **same label and an IOU >= 0.5** (Intersection over Union greater than 50%). This "match" is considered a true positive if that ground-truth object has not been already used (to avoid multiple detections of the same object).Third is confusion matrices for each dataset. Confusion matrices provide four types of results: true positives (TP), true negative (TN), false positive (FP), false negative (FN). True positive TP is the prediction in which the region inside the bounding box is a object class and our CNN model detect it as an object class. True negative TN is a case in which the region in the box is a not an object and detector also detects it as a no object. False positive FP is a detection in which the region inside the bounding box is detected as an object when there is no object in the box or there is actually an object inside the box but ground truth annotation does not recognize it. False negative FN are

those regions which are detected as background, but actually there is an object in it. Figure 5.5 shows the illustration of the confusion matrix.

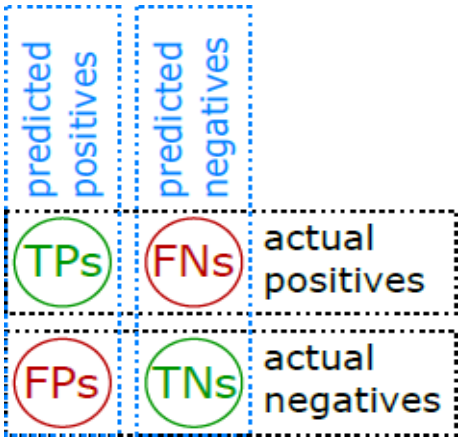

Figure 5.4: Confusion matrix representation [47]

Before training our yolov3 model on DOTA dataset, we need to prepare dataset for our model for efficient results. For this purpose, we convert the annotations of DOTA-v1.5 in yolo format i.e. " x_min,y_min,x_max,y_max,class_id " which our model take as input.Yolov3 model takes input images along with bounding box info and class labels from 0 to 15 to detect if there is an object in the given bounding box of the image or there is no object in that bounding box . To label our dataset according to YOLO labels we had to re-annotate our data. DOTA-v1.5 dataset annotations were in the form of bounding box location point of the objects present in the image. In the DOTA dataset, each instance's location is annotated by a quadrilateral bounding boxes, which can be denoted as "x1, y1, x2, y2, x3, y3, x4, y4" where (xi, yi) denotes the positions of the oriented bounding boxes' vertices in the image. Python was used to convert these annotations to simple Yolo annotations. A simple code was written to read the annotations of each image and generate the output yolo annotations by a simple formula. After a generation of annotations for all the images in DOTA-v1.5, the data set that was already divided into training, validation and testing sets by the ratio of 60 percent training, 20 percent validation and 20 percent testing data, we combine train and validation sets and then for data augmentation we use the image split technique with an overlap of 100.through a simple python code we split each image into a patch of

416x416 with an overlap of 100.this increase the images that are almost 1860 (by combining train and validation set) into 1 lac patches of equal width and height .Now we divide these images into train and validation set with ratio of 9:1.

Now the training set has almost 95000 images. To handle such large dataset annotations a text file was used that contains "image_file_path box1 box2 ... boxN" for all the images. For training we have the number of images used for training and validation with image height, image width and image depth according to tensor flow format. 3 channel input data with sixteen numbers of classes, batch size 4 and 90 epochs were used for the training process. Training, testing and validation arrays were generated for labels with shape: number of data. After completion of training process we save the trained model as HD file and obtain the training and validation loss for each epoch respectively.

Table 5.2 shows a comparison of different evaluation metrics from above model. It is clearly seen that precision and recall of some categories have been improved as compared to other techniques. .Figure 5.6 shows the confusion matrices for each dataset and their normalized forms as well. The miss rate calculated from confusion matrix of NWPU_10 dataset is better than the DOTA dataset for object detection inn aerial images. Figure 5.7 shows the training and validation loss of yolov3 model.

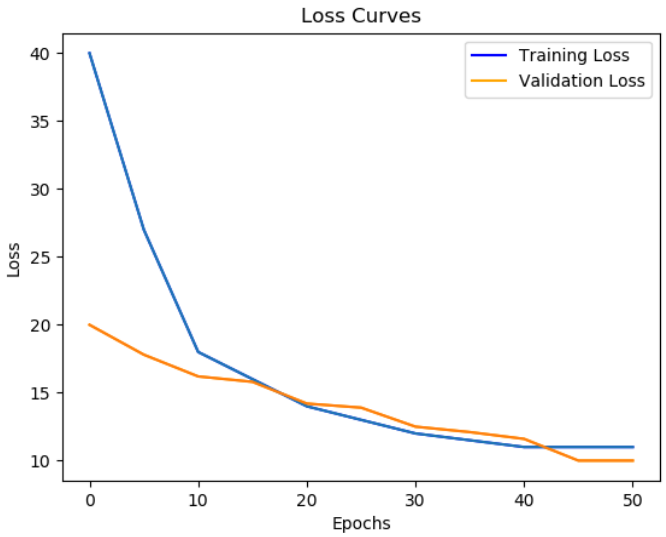

Figure 5.7: Training and Validation Loss curves of YOLOV3 model

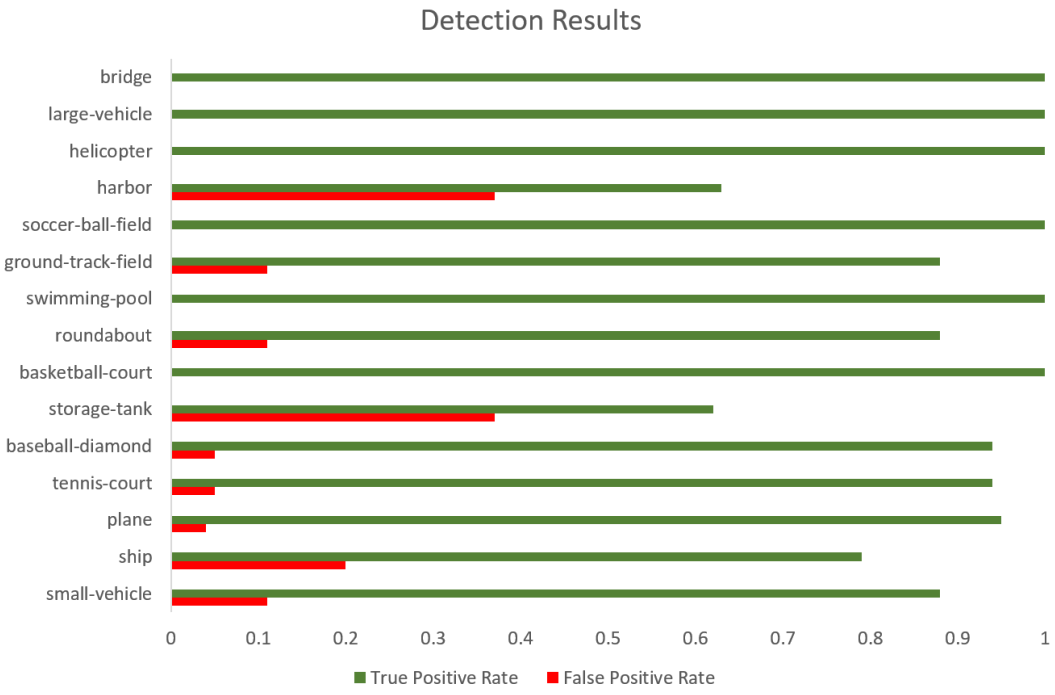

Figure 5.8: Detection Results

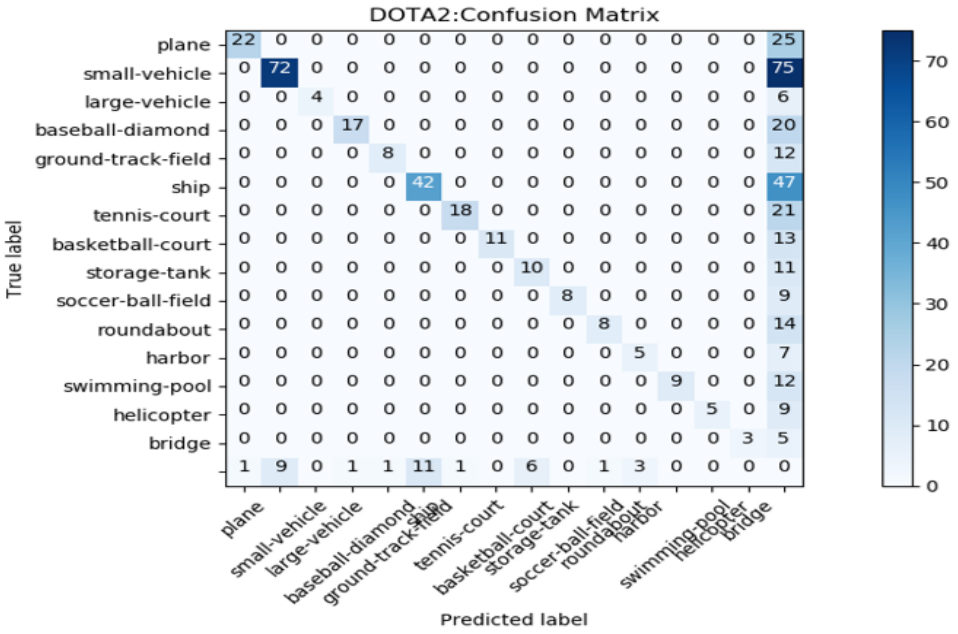

Figure 5.9: Confusion matrix for DOTA

Table 5.1: Evaluation Matrices: Numerical results (Precision, Recall) of baseline model evaluated with ground truths. The short names for categories are defined as: GTF–Ground field track, SV – Small vehicle, LV–Large vehicle, TC–Tennis court, ST–Storage tank, SBF–Soccer-ball field, RA–Roundabout and SP–Swimming pool

Classes	Plane	Bridge	SV	LV	ST	TC	GTF	SBF	RA	Harbor	Ship	SP
Precision	1.0	1.0	1.0	1.0	0.6	0.9	1.0	1.0	1	1.0	0.8	1
Recall	0.8	0.6	0.65	0.45	0.5	0.7	0.7	0.5	0.4	0.8	0.45	0.6

During the evaluation of trained model, we refer IOU that is intersection over union area.it is basically the overlap of actual and predicted bounding box of the objects. For example, if the value is≥0.5, it means that it is a positive example and if its value is <0.5, it is false example. When the IOU=0, it is a false negative example.

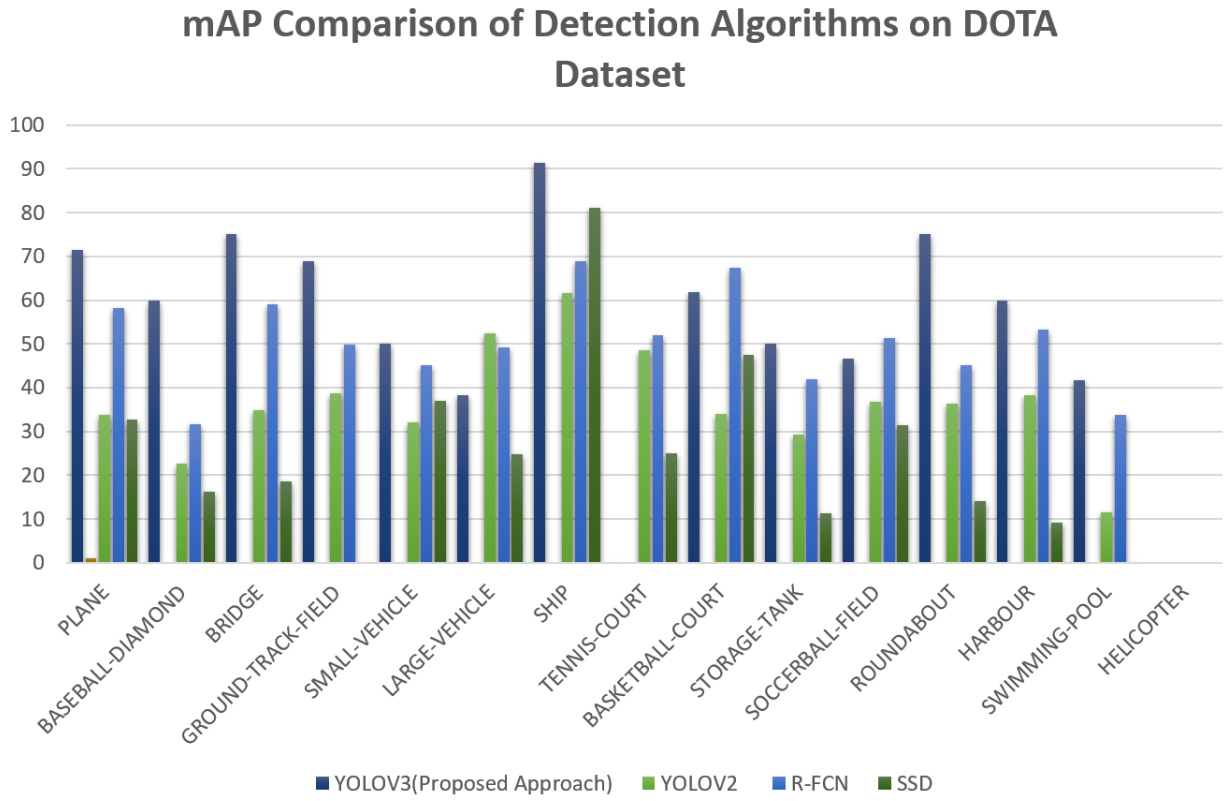

Figure 5.10: Map Comparison of detection algorithms on DOTA

The experimental results of proposed approach and other detection algorithms are shown in Table IV. The detection algorithm that is applied to aerial/satellite images for detection fifteen different types of objects, which have exceeds in both in speed and

accuracy. As compared to other techniques the AP value based on YOLOv3 reaches 61.17%, which completely demonstrates the precision gain of the YOLOv3 algorithm for goal detection. It is cleared from the above results that for real-time goal of detection, YOLOv3 algorithm is more suitable in terms of accuracy and speed. And if objects are congested and of various sizes then the results of YOLOv3 algorithm are better in such scenarios.

Chapter 6: CONCLUSION & FUTURE WORK

6.1 Conclusion

Object detection is a fundamental visual recognition problem in computer vision and has been widely studied in the past decades. Visual object detection aims to find objects of certain target classes with precise localization in a given image and assign each object instance a corresponding class label. Due to the tremendous successes of deep learning based image classification, object detection techniques using deep learning have been actively studied in recent years. In this research, an in-depth study of the dense YOLOv3 target detection model is proposed for detection and classification of objects in aerial / satellite images. From the evaluation and comparison of experimental results, the mean average accuracy of this technique is 61.17% and the average running speed is 21FPS which is much better than RCNN and other techniques. This model performs very well in detecting small and dense objects even in complicated and overcrowded aerial scenes. This technique significantly elevated the accuracy and operational efficiency. In addition, the detection technique proposed in this research can additionally be relevant to a large number of real time applications, however the only premise is that a giant quantity of data is required for training of detection model.

6.2 Contribution

- Review and Comparison of recent developments in object detection and localization systems using a convolutional neural network.
- Fully automated system for classification and localization of objects from aerial images for the purpose of safety and security.
- We trained and evaluated the model on DOTA dataset. The results show that our proposed network is really simple, fast and efficient. Both quantitative and qualitative comparisons of our network with the state-of-the-art networks are provided.

6.3 Future Work

The system proposed by us is quite efficient for detection and classification and provides good results for each instance present in the frame. This method avoids the problem of detecting small and dense objects efficiently. However the localization module can be improved by training the network on a diverse dataset. This system can be trained and modified a little to be used for detection of any kind of object from satellite and aerial images. There is a little overhead less mAp for some categories in localization instances due to less number of instances for some categories which can be minimized by using a large dataset that has large instance ratio for all categories.

REFERENCES

[1] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition",in: CVPR, 2016.

[2] R. Girshick, J. Donahue, T. Darrell, J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", in: CVPR, 2014.

[3] K. He, G. Gkioxari, P. Doll´ar, R. Girshick, Mask r-cnn, in: ICCV, 2017.

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, Semantic "image segmentation with deep convolutional nets and fully connected" crfs, in: arXiv preprint arXiv:1412.7062, 2014.

[5] Y. Sun, D. Liang, X. Wang, X. Tang, Deepid3: "Face recognition with very deep neural networks", in: arXiv preprint arXiv:1502.00873, 2015.

[6] J. Dai, Y. Li, K. He, J. Sun, R-fcn: "Object detection via region-based fully convolutional networks", in: NeurIPS, 2016.

[7] R. Girshick, "Fast r-cnn", in: ICCV, 2015.

[8] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: "Towards real-time object detection with region proposal networks", in: NeurIPS, 2015.

[9] Z. Cai, Q. Fan, R. S. Feris, N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection", in: ECCV, 2016.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C.Berg, "SSD: Single shot multibox detector", in: ECCV, 2016

[11] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, "You only look once:Unified, real-time object detection", in: CVPR, 2016.

[12] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Doll´ar, "Focal loss for dense object detection', in: ICCV, 2017.

[13] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition", in: Proceedings of the IEEE, 1998.

[14] [Online]. Available: http://deeplearning.net/tutorial/_images/mylenet.png.

[15] A. Krizhevsky, I. Sutskever, G. E. Hinton," Imagenet classification with deep convolutional neural networks", in: NeurIPS, 2012.

[16] K. Simonyan, A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, in: arXiv preprint arXiv:1409.1556, 2014.

[17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, “Going deeper with convolutions”, in: CVPR, 2015

[18] K. He, X. Zhang, S. Ren, J. Sun, “Identity mappings in deep residual networks, in: ECCV, Springer”, 2016

[19] [Online]. Available: https://github.com/KleinYuan/tf-object-detection.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[21] [Online]. Available: https://www.google.com/url?sa=i&source=images&cd=&cad=rja&uact =8&ved=2ahUKEwizqaK_s6HgAhUsy4UKHUDEB6YQjRx6BAgBEAU&url=https%3A%2F%

2Fwww.slideshare. net%2Fanirudhko ul%2Fsqueezing-deep-learning-into-mobile-phones&psig=AOv Vaw3aJKRT6z - wy1_Ly6Y57jIO&ust=1549346.

[22] [Online]. Available: https://www.google.com/ url?sa=i&source=images&cd= &cad= rja&uact =8&ved=2ahUKEwizqaK_s 6HgAhUsy4UKHUDEB6YQjRx6BAgBEAU

[23] Matthias Limmer,Hendrik Lensch, "Infrared Colorization Using Deep Convolutional Neural Networks," 2016.

[24] TeresaArauÂ jo, Guilherme Aresta, EduardoCastro, JoseÂ Rouco,Paulo Aguiar, Catarina Eloy, AntoÂ nio PoloÂ nia, AureÂ lio Campilho, "Classification of breast cancerhistology images using ConvolutionalNeuralNetworks," DOI: 10.1371/journal.pone.0177544, 2017

[25] Gustav Larsson, Michael Maire, Gregory Shakhnarovich, Learning Representations for Automatic Colorization, Computer Vision – ECCV 2016 Lecture Notes in Computer Science, 2016.

[26] R. Niessner, H. Schilling, B. Jutzi, "Investigations On The Potential Of Convolutional Neural Networks For Vehicle Classification Based On Rgb And Lidar Data," in ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2017.

[28] Ashnil Kumar, Jinman Kim, David Lyndon, Michael J. Fulham, David Dagan Feng, "An Ensemble of Fine-Tuned Convolutional Neural Networks for Medical Image Classification," IEEE J. Biomedical and Health Informatics 21 (1), pp. 31-40, 2017.

[29]Ronald Kemker, Carl Salvaggio, Christopher Kanan, "Algorithms for Semantic Segmentation of Multispectral Remote Sensing Imagery using Deep Learning," arXiv: 1703.06452v2 [cs.CV], , 2017.

[30] Jiwen Lu, Gang Wang, Weihong Deng, Pierre Moulin, Jie Zhou, "Multi-manifold deep metric learning for image set classification," in IEEE Conference on Computer Vision and Pattern Recognition, (CVPR), 2015.

[31] Simon Philipp Hohberg, "Wildfire Smoke Detection using Convolutional Neural Networks," 20 09 2015. [Online].

[32] Jin Kyu Kang, Hyung Gil Hong and Kang Ryoung Park, "Pedestrian Detection Based on Adaptive Selection of Visible Light or Far-Infrared Light Camera Image by Fuzzy Inference System and Convolutional Neural Network-Based Verification," 2017.

[33] Natalia Neverova, "Deep learning for human motion analysis," 2016. [Online].

[34] Samer Hijazi, Rishi Kumar, and Chris Rowen, "Using Convolutional Neural Networks for Image Recognition," in IP Group, Cadence.

[35] R. Girshick, J. Donahue, T. Darrell, J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation”, in: CVPR, 2014.

[36] Y. Freund, R. E. Schapire, et al., “Experiments with a new boosting algorithm”, in: ICML, 1996

[37] Li Zhuo· Liying Jiang , Ziqi Zhu1 , Jiafeng Li ,Jing Zhang ,Haixia Long, “Vehicle classification for large-scale traffic surveillance videos using Convolutional Neural Networks”, Springer-Verlag Berlin Heidelberg 2017.

[38] Yohei Koga , Hiroyuki Miyazaki and Ryosuke Shibasaki “A CNN-Based Method of Vehicle Detection from Aerial Images Using Hard Example Mining”, Remote Sens. 2018.

[39] Jun Sang, Zhongyuan Wu , Pei Guo , Haibo Hu , Hong Xiang , Qian Zhang and Bin Cai, “An Improved YOLOv2 for Vehicle Detection”, Sensors 2018.

[40] Tianyu Tang, Shilin Zhou, “Vehicle Detection in Aerial Images Based on Region Convolutional Neural Networks and Hard Negative Example Mining”, Sensors 2017

[41] Gui-Song Xia, Xiang Bai,Jian Ding, Zhen Zhu, Serge Belongie,Jiebo Luo, Mihai Datcu, Marcello Pelillo, Liangpei Zhang ,“DOTA: A Large-scale Dataset for Object Detection in Aerial Images”, , IEEE Conference on Computer Vision and Pattern Recognition, 2018

[42] Gong Cheng , Peicheng Zhou ,Junwei Han, “Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images”, IEEE Transactions on Geoscience and Remote Sensing , Dec. 2016

[43] Towards Data Science. (2018). *R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms*. [online] Available at: https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e [Accessed 6 Dec. 2019].

[44] Medium. (2018). *Real-time Object Detection with YOLO, YOLOv2 and now YOLOv3*. [online] Available at: https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2- 28b1b93e2088 [Accessed 30 Nov. 2019]

[45] A. F. J. Redmon, "“Yolo9000: better, faster, stronger”," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017

[46] J. Redmon and A. Farhadi, "“Yolov3: An incremental improvement," in CoRR, vol.abs/1804.02767, 2018

[47] Kdnuggets.com. (2019). *Object Detection and Image Classification with YOLO*. [online] Available at: https://www.kdnuggets.com/2019/09/object-detection-image-classification-yolo.html [Accessed 30 Nov. 2019]