CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD

Vehicle Detection and Classification in Aerial Images using Ensemble of Convolutional Neural Networks (CNN)

by

Mirza Zain Baig

A thesis submitted in partial fulfillment for the degree of Master of Science

in the

Faculty of Engineering
Department of Electrical Engineering

2023

I dedicate this work to my dearest parents, colleagues and friends

CERTIFICATE OF APPROVAL

Vehicle Detection and Classification in Aerial Images using Ensemble of Convolutional Neural Networks (CNN)

by

Mirza Zain Baig

(MEE191003)

THESIS EXAMINING COMMITTEE

S. No.	Examiner	Name	Organization
(a)	External Examiner	Dr. Adil Masood Siddiqui	MCS NUST, Islamabad
(b)	Internal Examiner	Dr. Aamer Iqbal Bhatti	CUST, Islamabad
(c)	Supervisor	Dr. Imtiaz Ahmad Taj	CUST, Islamabad

Dr. Imtiaz Ahmad Taj
Thesis Supervisor
April, 2023

Dr. Noor Muhammad Khan
Head
Dept. of Electrical Engineering
April, 2023

Dr. Imtiaz Ahmad Taj
Dean
Faculty of Engineering
April, 2023

Author's Declaration

I, **Mirza Zain Baig** hereby state that my MS thesis titled "**Vehicle Detection and Classification in Aerial Images using Ensemble of Convolutional Neural Networks (CNN)**" is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/abroad.

At any time if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my MS Degree.

(Mirza Zain Baig)
Registration No: MEE191003

Plagiarism Undertaking

I solemnly declare that research work presented in this thesis titled "**Vehicle Detection and Classification in Aerial Images using Ensemble of Convolutional Neural Networks (CNN)**" is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been dully acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS Degree, the University reserves the right to withdraw/revoke my MS degree and that HEC and the University have the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized work.

(Mirza Zain Baig)
Registration No: MEE191003

Acknowledgement

After being utmost grateful to **Almighty Allah** who gave me help and courage to complete my M.S research work, I would like to express huge gratitude towards the person who is not only my supervisor but also my mentor **Dr. Imtiaz Ahmad Taj** whose constant support, guidance and true motivation kept me steering in the right direction during my entire research work to get through this demanding and tiresome task.

I am grateful to my senior colleague **Mr. Asif Bhatti** and all colleagues at work place, who have encouraged and supported me in pursing MS degree, due to which I was able to spare time from busy work schedule for studies.

I would like to show my deepest gratitude and respect to my whole family, and particularly my parents, the ones to whom I owe all the success in my life. No words can express my gratitude to them, but I pray Almighty Allah to bless them and reward them.

(Mirza Zain Baig)

Abstract

The advent of low cost Unmanned Aerial vehicles (UAV) having on board imaging devices has led to a new field of aerial imaging. It has various applications including surveillance, intelligent transport management and security etc. One such application is to detect and classify automobiles in aerial images. In addition to general detection issues it poses challenges such as small target size, types of vehicles, illumination changes due to day and time, weather conditions, occlusions due to road side plantations, varying terrain, traffic conditions and other man made structures.

The vehicle detection on aerial images involves two basic tasks i.e., localization and classification. Combined structures of Convolutional Neural Networks (CNNs) specially designed to perform both these tasks simultaneously such as, You Only Look Once (YOLO) architectures, have shown great promise. However, simultaneous localization and classification approach by using a single network may be prone to errors as the optimization is achieved as a whole. In this thesis an alternate approach is explored, whereby localization and classification are taken as separate tasks and an attempt is made to achieve best results for each task individually through ensemble of networks.

YOLO based CNN architectures, are trained on existing aerial vehicle detection datasets (i.e. VAID and KIT-AIS), their results are analyzed and compared. Based on the results of these architectures for localization (bounding box estimation) and classification, two independent ensemble modeling approaches are studied. The first approach is to combine the bounding box outputs obtained from each of these networks, and the second ensemble is made on class predictions by these networks independently. After comparing the results with that of a single YOLO based network, a novel candidate box and class prediction ensemble method is proposed to improve the combined performance metrics. This proposed scheme is also validated on both the above mentioned datasets. The experimental results shows improvement in both localization and classification performance metrics and achieved 99.49% accuracy for classification and mean Average Precision (mAP) of 89.30% on VAID dataset and 98.80% accuracy for classification and mean Average Precision (mAP) of 56.63% on KIT-AIS dataset, which is decent improvement compared to the results of a single network.

Contents

List of Figures

List of Tables

Abbreviations

ANN	Artificial Neural Network
AP	Average Precision
COWC	Cars Overhead with Context
CNN	Convolutional Neural Network
CSP	Cross Stage Partial Network
DLR-MVDA	Multi-class Vehicle Detection and Orientation in Aerial Imagery
DOTA	Dataset of Object deTection in Aerial images
FP	False Positive
FN	False Negative
HoG	Histogram of Oriented Gradients
ICAO	International Civil Aviation Organization
IOU	Intersection of Union
KIT-AIS	Karlsruhe Institute of Technology - Aerial Image Sequence
KNN	K-Nearest Neighbor
ML	Machine Learning
mAP	mean Average Precision
OCBF	Overlapping Candidate Box Fusion
SIFT	Scale Invariant Feature Transform
SVM	Support Vector Machine
SCBF	Score Candidate Box Fusion
TP	True Positive
TN	True Negative
UAV	Unmanned Aerial Vehicle
VEDAI	VEhicle Detection in Aerial Imagery
VAID	Vehicle Aerial Image Dataset
VOC	Visual Object Classes
YOLO	You Only Look Once

Symbols

$\mathbf{Box}_k$	Candidate box with maximum detection score
$\mathbf{Box}_i$	Non maximum candidate box i
$\mathbf{C}_i$	Non maximum bounding box i after threshold
$\mathbf{M}_{th}$	Overlap threshold
$\mathbf{P}_i$	Detection score of box i
σ	Variance of Gaussian function
$\mathbf{w}_i$	Fusion weight i

Chapter 1

Introduction

1.1 Introduction

Object Detection is a computer vision technique for locating as well as label an object present in a image or video sequence, so it gives us the bounding box information (i.e bounding box co-ordinates) and its respective class. Object detection has long been a hot topic among researchers but lately it has become popular since the success of Deep Neural Networks (DNNs) and high computing resources available today.

An Unmanned Aerial Vehicles (UAVs) according to ICAO is a class of aircraft that is flown without a on-board pilot, by a operator from ground station. Over the last decade they have been utilized mostly by military of different countries particularly for surveillance, being a cost effective solution compared to traditional man based flights. The development of low cost quad-rotors as shown in figure 1.1, with embedded flight controller and on board high resolution cameras has surfaced way for a new kind of application called aerial imaging or photography. On the other hand military drone as shown in figure 1.2, is used particularly for surveillance and combat. Aerial imaging is basically taking pictures or recording video from cameras mounted on an air borne platform, which include quad rotors, UAVs etc. Of course, there are certain regulations that vary from country to country regarding payload, altitude, range etc which must be adhered to. It has various applications including surveillance, aerial photography, parking management, object tracking, security, search and rescue, food and supplies delivery etc. One of these application is automobile detection in which you can detect number of vehicles on a highway, manage traffic and send alert to commuters about potential traffic congestion or alert local area patrol regarding fatal accidents. All this can be accomplished through one person sitting inside control room and without having to deploy on ground manpower. Theses UAVs offer characteristics of being light weight, inexpensive and flexible making it easy for aerial imaging. They have range from 100m to 1Km and backup time from few minutes to couple of hours. Though to process all this huge volume of images is a hefty task and this is where computer vision can come to play. There are several challenges as well when comes to vehicle detection which are illumination changes, partial shading due to shadows of trees and other road side obstructions including sign boards, signals etc, small size of target to be detected, density of objects present and monotonic appearance.

FIGURE 1.1: General Purpose Drone
courtesy:nytimes.com/wirecutter/reviews/best-drones/amp/

FIGURE 1.2: MQ-9 Reaper Military Drone of USA courtesy:www.bbc.com/news/world-60047328/

1.2 Aerial Image Data

To understand and solve any problem, first of all you need analyze available data, and same is the case for a computer vision problem. In the context of vehicle detection in aerial images, these are sequence of labeled images (i.e defining a box around the object and its class). There are limited resources available in regard, and the available resources have issues like imprecise bounding boxes, small image sizes, and small number of categories available in these of datasets. The popular datasets in this domain are VAID, VEDAI, COWC, DLR-MVDA, DOTA and KIT-AIS datasets.

VEDAI dataset published by Jurie [4], contains around 1250 RGB and NIR images, having resolutions of 512x512 and 1024x1024. It has nine categories and around 3000 objects labelled in one of the nine classes. Annotations describe class, center point coordinates, direction and four corner point coordinates.

COWC data is collected over six distinct locations i.e. Toronto Canada, New Zealand, Germany, Columbus Ohio and Utah United States [5]. The images are standardized to 15 cm per pixel. The labeled images only mark the center point of a vehicle with a red dot. It does not provide the category or bounding box information.

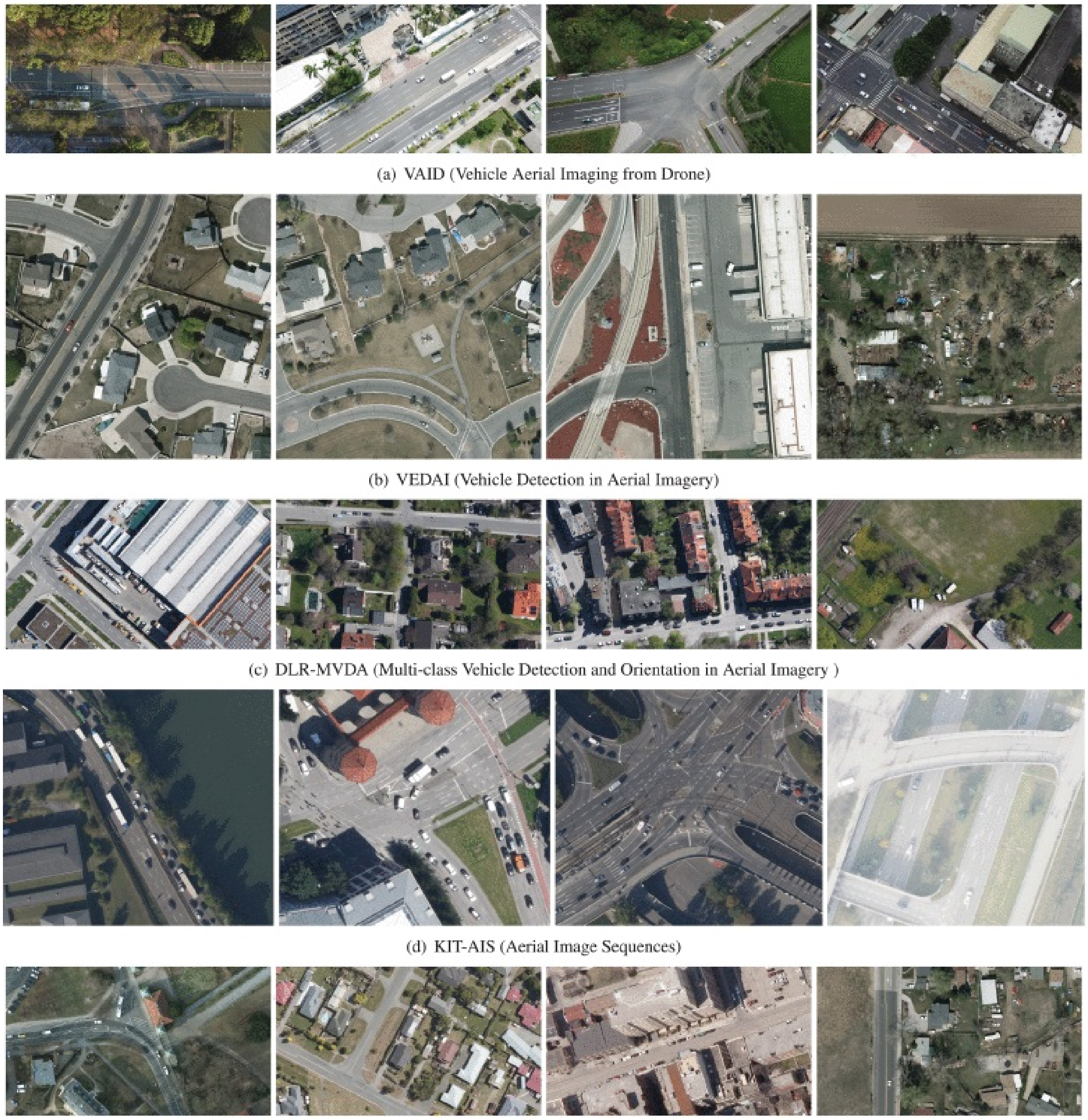
DOTA dataset has around 2800 images collected at resolution of 4000x4000 [6]. It contains 188K objects with different scales, orientations and labeled by quadrilaterals. This dataset is for general purpose use with only two vehicle categories out of 15.

KIT-AIS is another similar kind of dataset available for vehicle recognition research published by researchers of Karlsruhe Institute of Technology, Germany. The vehicles in this dataset, pose challenges like small, target size, exhibit variability such as multiple orientations and occlusions etc. The annotation format is YOLO unlike pascal voc as in VAID dataset. It contains four different types of vehicles namely car, truck, bus, and minibus. It has approximately 157 images having resolutions of 1080 x 1920, 4000 x 4000 and 1280 x 720 pixels captured by small aircraft at an height of about 300m.

VAID (Vehicle Aerial Imaging from Drone) [1], is the latest addition to these datasets. The authors have gathered over 6,000 aerial images from various locations in Taiwan under varying lighting and viewing angle situations through hours of video recording using quad copter with bottom mounted camera. The images in this dataset were captured by a DJI Mavic Pro drone while it was capturing video at a height of roughly 90 to 95 meters. The output resolution is 2720x1530 pixels and about 24 frames per second. A typical vehicle size in the image, which is approximately 110x45 pixels and has a length of 5m and a width of 2.6m. Later, the photographs are resized to a resolution of 1137x640, and a sedan is around the size of 20x40 pixels in the image. Taiwan is a nation in East Asia also called a colony of China. It is situated at the junction of the East and South China Seas in the northwest Pacific Ocean, with Japan to the northeast, Philippines to the south, and the People's Republic of China (PRC) to the northwest. Taiwan is known for technological hub with large scale manufacturing of electronic goods. Taiwan is also known for mountain ranges dominating the eastern two-thirds and plains in the western third, where the majority of the country's densely populated cities are located. The capital of taiwan is Taipei. Taiwan has a population of 24 Million and is also among the densely populated countries in the world. This dataset includes diverse traffic and road conditions and spans ten distinct geographic areas in southern Taiwan. Images are taken on bright days when there is enough light, in the afternoon, and in the evening when there is less light for imaging. These images depict three different geographical locations: a metropolitan area, a suburban area, and a college campus. The images were converted to JPEG format and contains seven different types of vehicles namely sedans, minibuses, trucks, pickups, buses, cement trucks, and trailers. The software which was utilized to annotate these images is LabelImg tool, and the output format of annotated images is PASCAL VOC, which includes the corresponding bounding box coordinates as well as class label. Figure 1.4 shows an image from each vehicle category. VAID dataset will be used as a benchmark dataset for training of CNN networks for proposed method. Table 1.1 shows the summary of aerial image datasets and figure 1.3 shows few images from each datasets, table 1.2 shows scene wise distribution of vehicle categories in images for this dataset.

Table 1.1: Breakup of Available Datasets [1]

Title	**Images**	**Resolution**	**Scale**	**Automobile Size**
VEDAI	1250	512x512	25cm	10x20
COWC	53	2000x2000	15cm	24x48
DLR-MVDA	20	5616x3744	13cm	20x40
KIT-AIS	241	300-1800	12.5cm-18cm	15x25
VAID	5985	1137x640	12.5cm	20x40



(a) VAID (Vehicle Aerial Imaging from Drone)

(b) VEDAI (Vehicle Detection in Aerial Imagery)

(c) DLR-MVDA (Multi-class Vehicle Detection and Orientation in Aerial Imagery)

(d) KIT-AIS (Aerial Image Sequences)

(e) COWC (Cars Overhead With Context)

Figure 1.3: Few Images from Each Dataset [1]

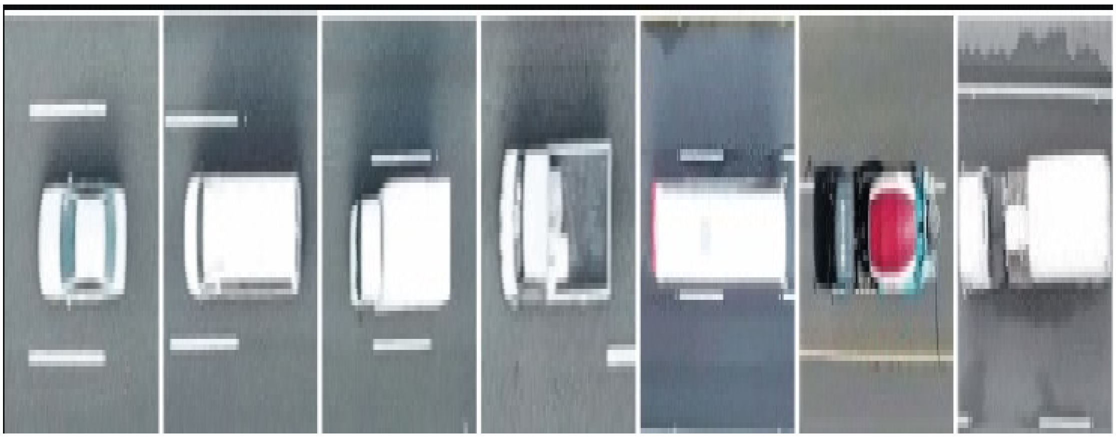


Figure 1.4: Each Type of Vehicle in VAID Dataset [1]

Table 1.2: Scene Wise Vehicle Categories Distribution [1]

	Images	**Sedan**	**Minibus**	**Truck**	**Pickup**	**Bus**	**Cement Truck**	**Trailer**
University Campus	3527	11385	406	605	611	292	17	36
Urban Area	1118	18349	95	1014	822	225	6	10
Suburb	1610	10596	0	1568	1578	63	168	758
Total	5985	40330	501	3187	3011	580	191	804

1.3 Object Detection Techniques

There are two approaches to this task, the traditional machine learning based techniques which uses some sort of features like edges, corners, textures, gradient orientation or colors, and from these features distinct features are selected which then utilized to train a classifier, the feature extraction is done by machine learning engineer or data scientist having domain specific knowledge. Then using a sliding window operator over the image to categorize the positive and negative samples. Therefore this is a three stage detection system which is computationally extensive at times.

On the other hand, neural network based approaches, which are frequently based on convolutional neural networks (CNN), Convolutional neural networks (CNNs), which have been found to have significant improvement on object detection and classification in the past few years. These are spatially connected artificial neural networks (ANN) having many hidden layers, due to which called deep neural networks. They work by convolving filters of different sizes and generate feature maps to perform training, sequentially reducing size of feature maps using pooling layer, until they are small enough to be passed to fully connected layer followed by softmax function in the end to get final class, as well as they also have a regression layer used to perform bounding box regression during training and estimating bounding box coordinates.

1.4 Ensemble Modeling

Ensemble modeling is used in machine learning that combines the predictions from multiple models to seek better performance. In object detection ensemble technique can be applied in two ways, one is to combine the redundant/overlapping bounding boxes information in a way that the final bounding box is more closer to its target position. The second way could be used find the correct class of the object using class prediction ensemble, in this thesis we propose an ensemble based method for both the detection and classification.

1.5 Thesis Overview

Chapter 1 Presents a brief introduction to aerial imaging in UAVs, its various applications and challenges involved. The benchmark datasets available in this domain, and existing techniques for the said task, and how we want to approach the said task using model ensemble. **Chapter 2** The second chapter of this thesis presents the literature review of existing machine learning, and CNN based approaches are explained. Afterward, comparative analysis of existing CNN based architectures by some researchers is also discussed, followed by how to check performance of these networks is shown. Research gap analysis and problem statement are highlighted in light of the literature survey. **Chapter 3** The detection ensemble scheme of trained networks is discussed, performed and evaluated along with results of individual CNN is discussed in detail. **Chapter 4** The classification ensemble of trained networks is discussed, performed and evaluated along with results of individual CNN is discussed in detail. **Chapter 5** The fifth chapter of this thesis work presents the results and discussion section in which comparison between known studies is made. Classification and detection performance analysis of all competing techniques is made, and based on the results a novel ensemble model for both localization and classification is proposed. **Chapter 6** presents the conclusion section with the future work. This section concludes that the proposed technique shows better performance as compared to single CNN network.

1.6 Chapter Summary

This chapter gives brief overview of object detection in aerial images through UAVs. The available data in this domain and the challenges involved in vehicle detection are discussed. Ensemble method in object detection is discussed.

Chapter 2

Literature Review

2.1 Classic Machine Learning Based Methods

The traditional machine learning based approaches such as Scale Invariant Feature Transform (SIFT), Histogram of oriented Gradients (HOG) etc. used for object detection involves a three stage detection system in which first stage is to frame candidate regions and locate objects using sliding window, Second stage involves extracting features from these candidate regions using HOG, SIFT and third stage is to train a classifier for classification like SVM etc. So, using a sliding window operator of fixed size and generating features or descriptors, which is a N-dimensional vector with an associated class. A database is formed based on these features extracted, data is split into training and testing data, a classifier is trained on this data like K-Nearest Neighbor (KNN), or a Support Vector Machine (SVM) etc. This is also referred to as supervised learning. The model generated is then tested on test data to evaluate performance.

The difficulty with this conventional technique is that computer vision engineer has to decide which features to choose from, and as the number of classes increases, feature extraction becomes more difficult. Moreover multiple sliding windows are required to capture objects of different sizes, also which is computationally extensive and not suitable for real time applications. Similar to the above mentioned scheme is utilized by Shao [7], here they have used multiple low level features i.e shape, texture and color followed by training a SVM based classifier. After training detection is performed by exhaustive search and repetitive detections are eliminated by non-maximum suppression in post processing step. They have applied their algorithm on vaihingen dataset and showed promising results. There are few limitations to Shao's work, firstly they have used three different kinds of features i.e Histogram of Oriented Gradient (HoG), Local Binary Pattern (LBF) and RGB opponent Histogram, although all these low level features but still they are computation extensive, and second there work is limited to single category i.e car, so its basically just detection and not classification combined.

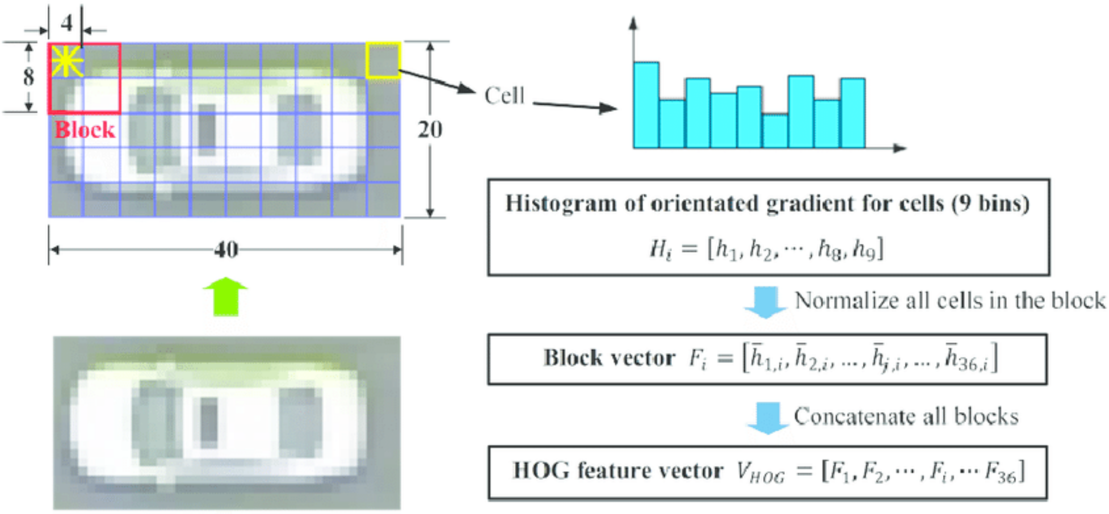

4
8
Block
20
40
Cell
Histogram of orientated gradient for cells (9 bins)
$H_i = [h_1, h_2, \cdots, h_8, h_9]$
Normalize all cells in the block
Block vector $F_i = [\bar{h}_{1,i}, \bar{h}_{2,i}, \ldots, \bar{h}_{j,i}, \ldots, \bar{h}_{36,i}]$
Concatenate all blocks
HOG feature vector $V_{HOG} = [F_1, F_2, \cdots, F_i, \cdots F_{36}]$

Figure 2.1: Process Flowchart HoG [2]

Another approach could be to simply compute optical flow or apply background subtraction which is based on the subtraction of consecutive frames followed by applying a threshold and connected components labeling to detect moving vehicles, and then passing the detected vehicle to a classifier to find its class as shown figure 2.2. To account for the video stabilization issues due to six Degree of freedom (DOF) motion of a helicopter, a frame by frame video registration scheme using lucas Kanade (KLT) based feature tracker to automatically determine control-point correspondences is proposed [8]. This scheme is although fast limited to moving vehicles only.

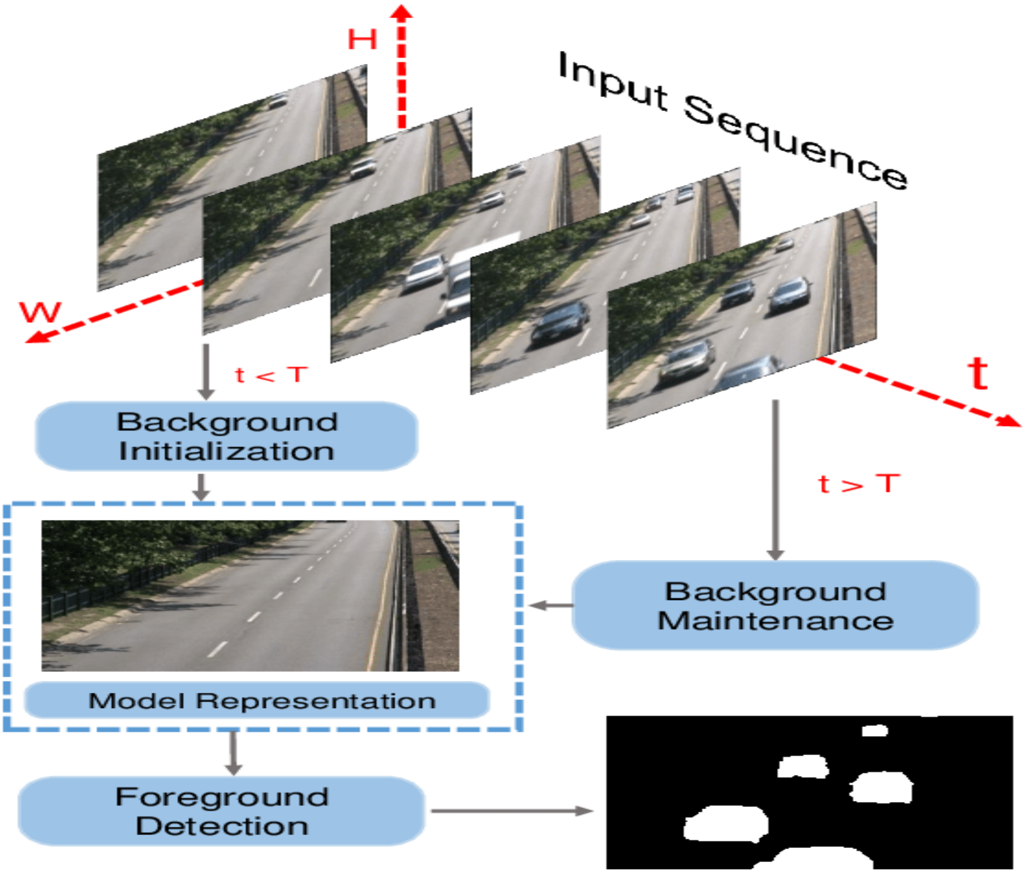
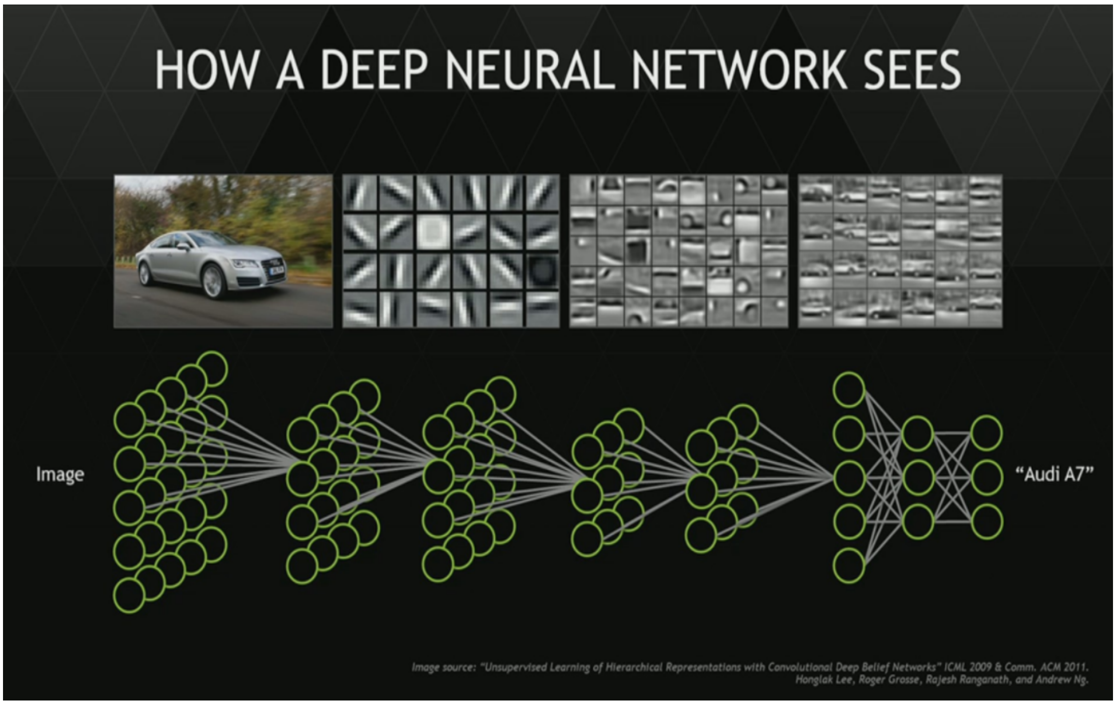


FIGURE 2.2: Background Subtraction Process [3]

2.2 Modern Deep Learning Based Techniques

The modern deep learning based approach consists of Convolutional Neural Network (CNN's), which is the current state of art in object detection, there are two kinds of networks in CNN's specifically designed to deal with images, the one stage You Only Look Once (YOLO) based architectures and two stage RCNN based architectures, and unlike artificial neural network (ANN) which are fully connected networks, the CNN's are spatially connected to the network because of the big image size like 320x320, 640x640 etc. These networks can have multiple hidden layers and can have hundred to millions of weight parameters, and therefore so called deep neural networks. The RCNN based architectures first extracts region proposals using an algorithm such as selective search before passing it to the network and therefore it is essentially a image classifier, on the other hand YOLO based architecture outputs four additional numbers i.e bounding box coordinates apart from class prediction. CNN work by applying filters of different sizes extending the full depth of input, it is equivalent to sliding window operation in conventional machine learning discussed earlier. The resultant outputs are feature maps which are passed to some activation function usually ReLU activation function and then combined with pooling layers to reduce their size until they are small enough to be passed to fully connected network. In this way CNN perform kind of pattern recognition, and by adding layer after layer we end up learning hierarchical features, where initial layers are low level features like edges, corners etc., and later layers having high level features like blobs etc. In the end there is a full connected layer followed by soft max function to get final class. These filters have weight parameters that are learned by network during training, The hierarchical structure of CNN is shown in figure 2.3.



HOW A DEEP NEURAL NETWORK SEES

Image source: "Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks" ICML 2009 & Comm. ACM 2011. Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Ng.

FIGURE 2.3: Hierarchical Features in Convolutional Neural Network (CNN)

Girshick in 2015 proposed an improvement in existing R-CNN architecture [9], to improve training and testing speed while also increasing detection accuracy. The entire image and a number of object proposals are input into a Fast R-CNN network. The network creates a convolutional feature map by first processing the entire image with a number of filters and max pooling layers. A region of interest (RoI) pooling layer then extracts a fixed-length feature vector from the feature map for each object proposal. Each feature vector is fed into a series of fully connected (fc) layers that eventually branch into two sibling output layers, one layer that outputs four real-valued numbers for each of the K object classes and another layer that generates softmax probability estimates over K object classes plus a "background" class. For one of the K classes, each pair of 4 values represents the bounding-box coordinates.

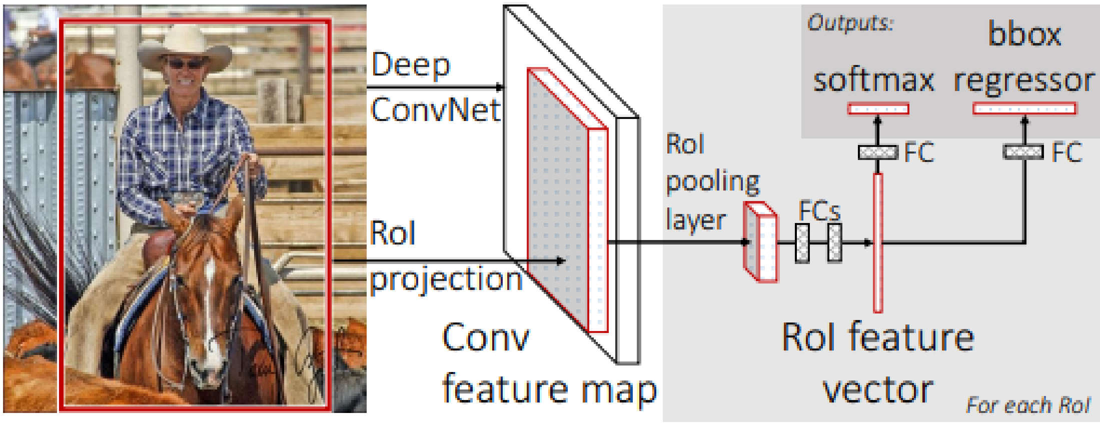


FIGURE 2.4: Fast R-CNN Architecture [9]

Shaoqing and Girshick added the Region Proposal Network (RPN) [10], which shares full-image convolutional features with the detection network, as yet another enhancement to the existing RCNN design in 2016. A fully convolutional network known as an RPN forecasts object bounds and scores at each location at the same time. The RPN is completely taught to produce excellent region proposals that are used for detection. By combining RPN and Fast R-CNN into a single network and utilizing each other's convolutional features, they have considerably speed up computation. The RPN can be trained from beginning to end especially for the job of producing detection proposals, making it a type of fully convolutional network (FCN). RPN's may be completely taught for the purpose of producing detection proposals. RPN's are made to forecast region proposals with a variety of scales and aspect ratios effectively. They present novel "anchor" boxes that serve as references at various scales and aspect ratios as an alternative to conventional techniques that use pyramids of images.

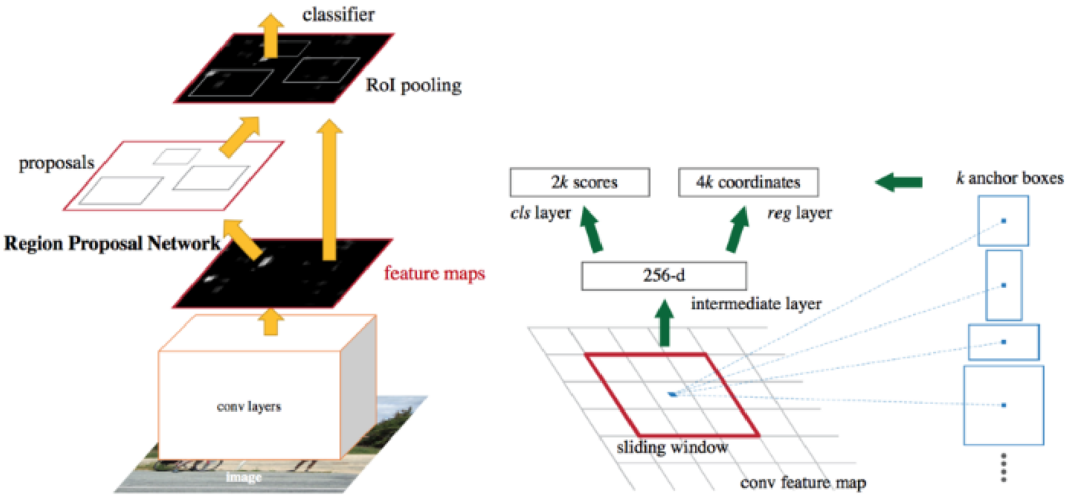


Figure 2.5: Faster R-CNN Architecture [10]

You Only Look Once (YOLO) architecture was first introduced in 2016 by Joseph Redmon [11]. Their method is completely different. A single neural network is used to process the entire image. Bounding boxes and probabilities for each region are predicted by this network after it divides the image into regions. The predicted probabilities are used to weight these bounding boxes. Bounding boxes and class probabilities are predicted by a single neural network from complete images in a single evaluation. Since the entire detection pipeline consists of a single network, detection performance can be optimized from beginning to finish. The real-time, 45 frame-per-second image processing speed of the YOLO architecture makes it incredibly quick.

YOLO make fewer false positive predictions on background, but YOLO makes more localization mistakes. YOLO picks up very broad representations of things.

It beats alternative detection strategies. The GoogLeNet model for image classification served as the model of reference for its network design. 24 convolutional layers precede 2 completely connected layers in yolo network. They use 1 x 1 reduction layers followed by 3 x 3 convolutional layers. The full network is shown in Figure 2.6 For evaluating YOLO, final prediction is a 7 × 7 × 30 tensor.

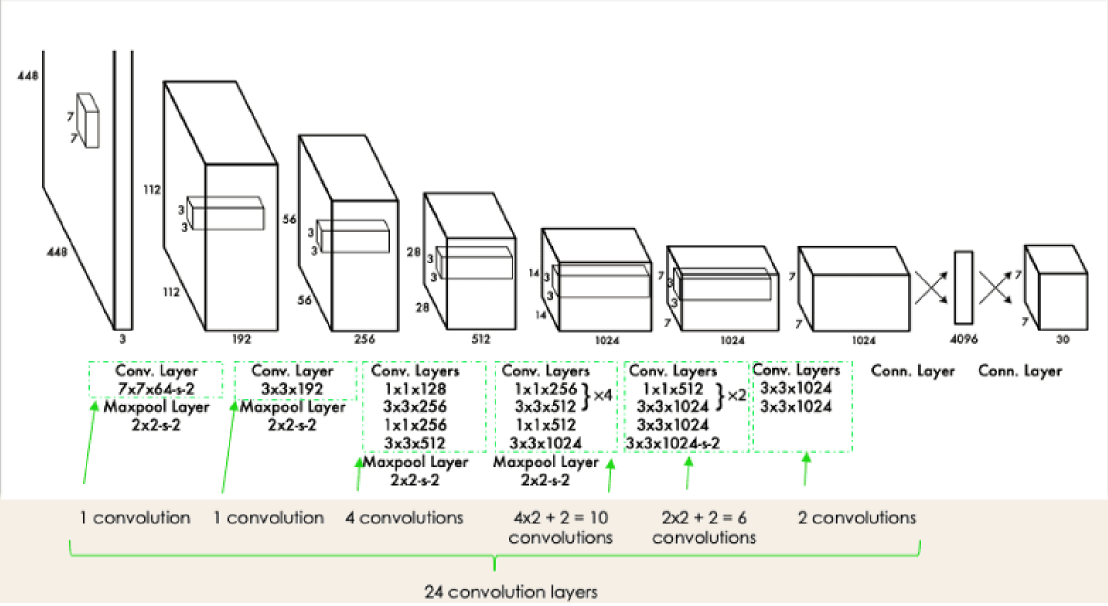


FIGURE 2.6: YOLO Architecture [11]

2.3 Comparative Analysis

Some of the researchers have performed comparative analysis of both YOLO and RCNN based architectures with application to detection in aerial images. In 2017, Sommer used the DLR 3K Munich Vehicle Aerial Image Dataset and the Vehicle Detection in Aerial Imagery (VEDAI) dataset to assess the potential of Fast R-CNN and Faster R-CNN for aerial images [12]. They tested eight various object proposal techniques, including Selective Search (SS), Edge Boxes (EB), and RPN, to produce a set of candidate regions. By adjusting the RPN's anchor boxes and the output resolution of the final convolutional layer used as a feature map to account for the lower spatial resolution of the aerial imagery and the resulting smaller object sizes, they demonstrated that RPN clearly outperforms all other proposed methods.

Another researcher compared Faster RCNN with YOLOv3 in context of car detection in aerial images [13]. It is observed that although both algorithms have high precision rates (99.66% for Faster R-CNN and 99.73% for YOLOv3), when comparing recall, we find that YOLOv3 performs much better than Faster R-CNN (79.40% for Faster R-CNN and 99.07% for YOLOv3). YOLOv3 takes 57 ms on average for one image. The average processing time for each image using Faster R-CNN is 1.39 seconds. They claimed that YOLO network outperformed faster RCNN in terms of speed and accuracy and the reason behind it is that there are couple of improvements in YOLOv3 archietecture. The use of the multi-label classification, as opposed to the mutually exclusive labeling used in earlier iterations, is the first advancement made with YOLOv3. It determines the likelihood that an item belongs to a particular label using a logistic classifier. Previous iterations generated scores using the soft max function. It substitutes the binary cross-entropy loss for each label for the general mean square error used in the earlier versions for the classification loss. The use of various bounding box predictions is the second enhancement. It links the bounding box anchor with the score value of 1 when that anchor overlaps a ground truth object more than others. Other anchors that overlap the ground truth object by a specified threshold or more are ignored. (0.7 is used in the implementation). As a result, YOLOv3 gives each ground truth object a single bounding box point. Utilizing feature pyramid networks for cross-scale prediction is the third advancement achieved. Boxes are predicted by YOLOv3 at three distinct scales, from which features are then extracted. A 3-d tensor that represents bounding box, score, and prediction over classes is the prediction outcome of the network. YOLO feature extractor, known as Darknet-53, and is so called because it has 53 layers. It is the fifth enhancement and employs a skip connections network with 53 layers that was modeled after ResNet. Additionally, 3 x 3 and 1 x 1 convolutional layers are used. It demonstrated state-of-the-art accuracy while using less floating point calculations and moving more quickly.

2.4 Ensembling Techniques

As YOLO splits the image into cells so we might get multiple redundant detection for a test object if the object is spread over multiple cells, and now we have perform some post processing to filter out the best candidate box among thes redundant boxes. The candidate box selection method is a crucial part, as its performance will directly affect the accuracy. Even if the neural network can detect many perfect bounding boxes, if there is no good candidate box selection method to filter them out, there will be a phenomenon of missed detection and false detection. Numerous methods primarily employ non-maximum suppression techniques. Some of the non-maximum suppression techniques include the conventional NMS, soft NMS and its variants.

2.5 Non-Maximum Suppression

For a particular object in test image we take and combine all the outputs from each network and simply pick the highest score bounding box and suppresses the other bounding boxes with non-maximum scores based on overlapping threshold (IoU), as described in equation 2.1, and repeat the process for the rest of the objects in the image. Conventional NMS technique is based on single level decision boundary and due to which a lot of boxes get eliminated resulting in lower detection accuracy.

$$\begin{gathered} Pi = \{Pi, iou(Box_k, Box_i) < M_{th} \\ 0, iou(Box_k, Box_i) \geq M_{th}\} \end{gathered} \tag{2.1}$$

where The bounding box Box_i has class probability score P_i. Box_k has the maximum class probability of any bounding box. The non-maximum bounding box is called Box_i. The overlap threshold is M_{th}. $iou(Box_k, Box_i)$ is an equation that represents the overlap of two bounding boxes as follows,

$$iou = \frac{Box_k \cap Box_i}{Box_k \cup Box_i} \tag{2.2}$$

2.6 Soft Non-Maximum Suppression

In the soft maximum suppression we reduce the detection score of overlapping candidate boxes gradually by imposing a penalty on them and therefore decreasing the chance to miss detection as mentioned in Eq. 2.3 and Eq. 2.4. The fixed threshold method has difficulty to adjust in complicated environments, as can be observed in the conventional NMS method. As a result, two Soft-NMS approaches based on penalty factors were suggested in the literature [14]. This approach steadily decreases the number of candidates by lowering the detection score.

$$\begin{aligned} Pi &= \{Pi, iou(Box_k, Box_i) < M_{th} \\ Pi &= \{Pi * (1 - iou(Box_k, Box_i)), iou(Box_k, Box_i) \geq M_{th} \end{aligned} \tag{2.3}$$

$$\begin{aligned} Pi &= \{Pi, iou(Box_k, Box_i) < M_{th} \\ Pi &= \{Pi * e^{\frac{-iou(Box_k, Box_i)^2}{\sigma}}, iou(Box_k, Box_i) \geq M_{th} \end{aligned} \tag{2.4}$$

where sigma represents the variance of the Gaussian function.

Equation (2.3) is a linear penalty function, and Equation (2.4) is a Gaussian penalty function. The Soft-NMS approach simply lowers the detection score of bounding boxes with high overlap. This technique can, in part, lessen the likelihood of missing detection. The bounding box list will abruptly alter when the overlap degree is close to the threshold value M_{th} because the linear penalty function curve is not continuous. The ideal penalty function should be a continuous smooth curve, where the detection score smoothly declines as the bounding box overlap grows. In light of this, another technique that helps the Soft-NMS method's penalty function also called softer NMS.

The Softer NMS proposed in [14], whose penalty function as follows,

$$Pi = Pi * (1 - iou(Box_k, Box_i))^3 \tag{2.5}$$

According to the NMS technique and the modified NMS method, the main consideration in determining whether to keep a bounding box is its detection score exceeds a specific pre-defined threshold or whether its overlap IOU value exceeds the pre-define threshold. Soft NMS technique performs better than due to the fact that it reduces the detection score gradually to avoid overlapping boxes being drastically deleted and causing a miss detection. It is observed that non linear penalty term introduced as described in Eq. 2.5, definitely improves performance, because now we are not directly eliminating bounding boxes but picking the highest detection box gradually.

2.7 Performance Evaluation

Performance metrics are used to evaluate how if the model is good enough to be deployed and can be used for the application it is intended for, the model is definitely evaluated on test data. There are two kinds of performance metrics used i.e classification and detection. In classification metric tells us how well our model is recognizing the object of interest. Detection metric tells us how well our model is localizing our object of interest in terms of bounding box coordinates. In classification, performance metric is usually accuracy which is proportion of true positives over sum of all predictions. The other metrics include precision, recall and F1 score. The precision tell us how many positive predictions are actually positives. Recall tells us that how many predicted positives are correctly classified. To evaluate detection Intersection of Union (IOU) is used, which quantifies how close are the ground truth bounding box and predicted bounding box. IOU measures the overlap between ground truth box and predicted box over their union.

The predictions made by the detection network is again categorized into four categories True Positives (TP), False Positives (FP), False Negatives (FN), and True Negative (TN) although TN is not used as it describes the scenario in which empty boxes are appropriately identified as being without an object. In this case, it would be clear that the model would identify hundreds of empty boxes, adding

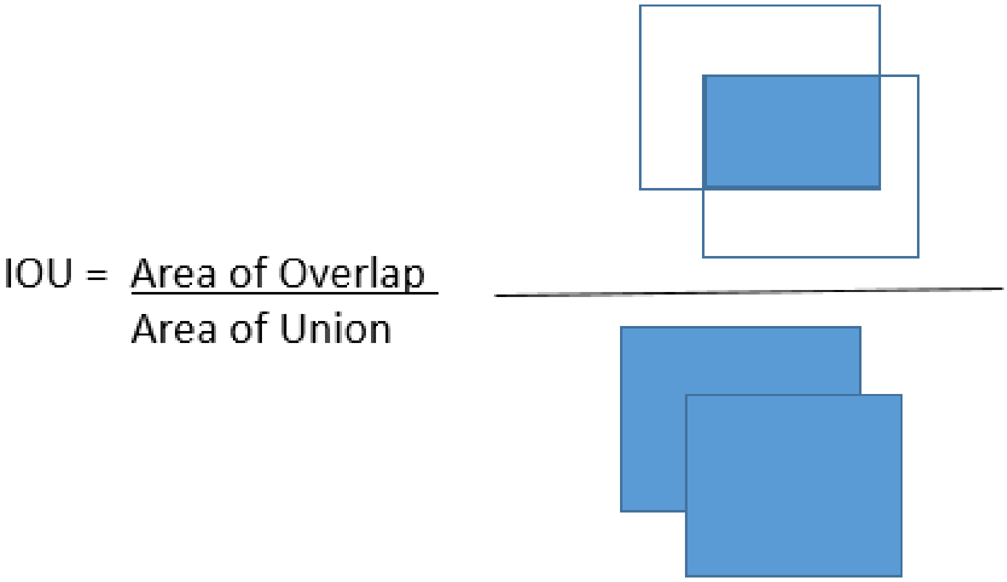


Figure 2.7: Intersection of Union

very little to no benefit to our algorithm. Precision, also known as the positive predictive value, is the likelihood that predicted bounding boxes will match the actual ground truth boxes. Recall, which is also known as sensitivity, estimates the likelihood that ground truth objects will be successfully detected. Average precision (AP), a single number metric that combines precision and recall and summarizes the Precision-Recall curve by average precision across recall values from 0 to 1, is used to assess the performance of object detectors. We interpolate the matching precision for a specific recall value r by taking the maximum precision at a set of 11 spaced recal points (0, 0.1, 0.2,..,.., 1) where AP averages precision. Take the maximum accuracy point where its associated recall value is to the right of r, to put it another way, Mean Average Precision (mAP) averages AP over the N classes in this situation, where Precision is interpolated at 11 recall levels, hence the name 11-point interpolated average precision.

2.8 Gap Analysis

The following key areas are identified after review of literature in this domain

- The conventional machine learning based approaches for object detection particularly in aerial images relied on either limited to moving objects or hand crafted feature extraction (i.e HoG, LBP, color, texture, SIFT etc.) has to be done followed by a sliding window based classifier such as Support Vector Machine (SVM) to label potential regions in image one by one, this is computational extensive and thus slow.
- The current state of the art deep learning based approaches are comparatively fast but then there is lack of studies exploring separate optimization for Region of Interest (ROI),and simultaneous localization and classification approach by using a single CNN may be prone to errors as the optimization is achieved as a whole in a multi-class scenario.
- There is also lack of comparative analysis between different CNN based object detection architectures due to the fact that these networks are still pretty much new and is a hot topic among researchers.
- There is also limited work done on ensemble modeling techniques to define a framework, to combine the outputs of multiple object detection networks together to increase overall accuracy and performance.

2.9 Problem Statement

Conventional machine learning based techniques for object detection in aerial images exist but are slow and less accurate, while deep learning based approaches are fast but they are not completely optimized for localization and classification in a multi-class problem, there is also limited work done on ensembling techniques to combine the outputs of multiple object detection networks together with the intent to increase performance.

2.10 Chapter Summary

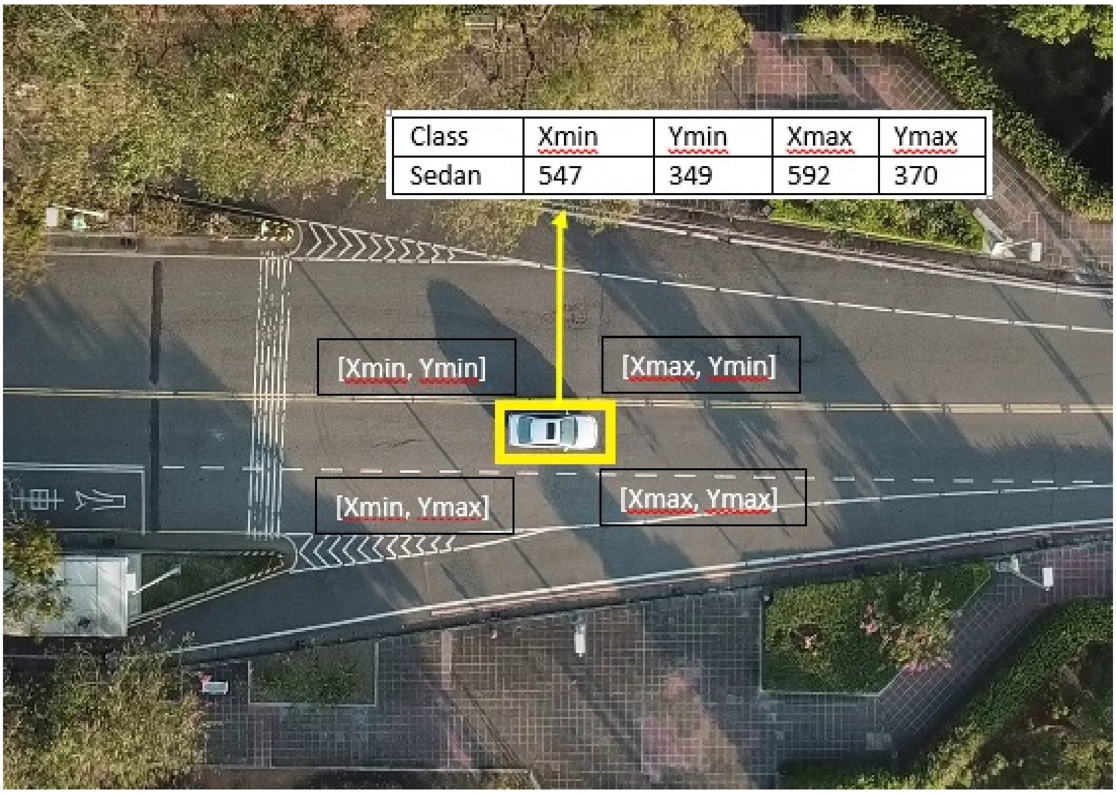
This chapter briefly describes the conventional machine learning based and current state of art deep learning based approaches used in object detection. Then we described the RCNN and YOLO based CNN architectures used for this task, the need for non maximum suppression to remove redundant boxes and keep boxes with highest overlap. We also discussed the ensembling techniques criteria to evaluate these object detection networks and key performance metrics that are utilized for assessment of these networks. After literature reviewing, analysis is made and highlight the ares where gap is present. In the end define problem statement and how we approach the said problem.

Chapter 3

Localization Ensemble

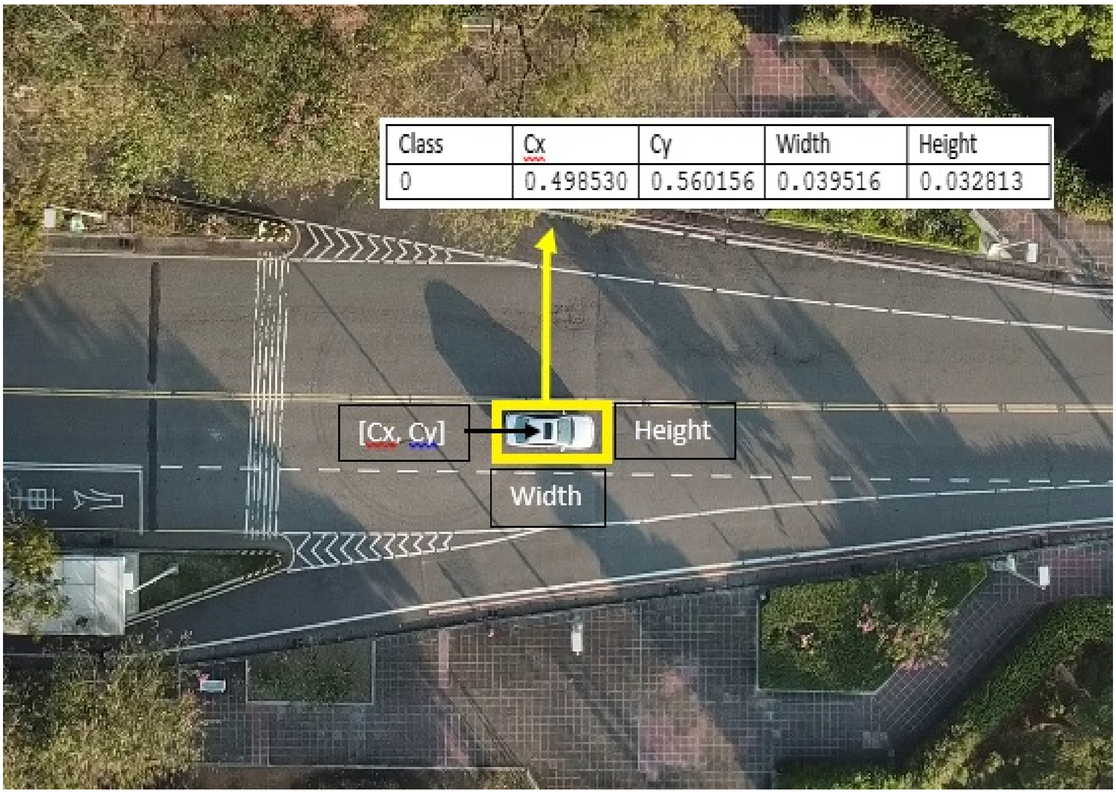
3.1 Training YOLO Networks

We trained and analyzed multiple YOLO based CNN networks through transfer learning approach, in which we choose an pre-trained model i.e a model trained on a similar problem in order to gauge the gained knowledge, change its output and input layers according our specific requirements i.e image size, no of classes etc., and then retrain the model. This way we are using already stored knowledge of a related problem and applying it another problem and hence given name transfer learning. Before we can train any network first we have to prepare the dataset in format that is accepted by the network to be trained, we resize the image size according to the network input size and re-scale the annotations accordingly. Therefore format conversion had to be done on the labeled ground truth data to be able to train these networks. The VAID had annotation in PASCAL VOC and KIT-AIS dataset had annotations in YOLO, annotation format had to be converted into YOLO to train these datasets on YOLO architectures, as well as image resizing to make it compatible with input image size of network to be trained.



Class	Xmin	Ymin	Xmax	Ymax
Sedan	547	349	592	370

FIGURE 3.1: PASCAL VOC Annotation in VAID Dataset Image Name 000094



Class	Cx	Cy	Width	Height
0	0.498530	0.560156	0.039516	0.032813

FIGURE 3.2: YOLO Annotation Format in VAID Dataset Image Name 000094

The next step before training is to prepare both VAID and KIT-AIS dataset i.e. to split datasets into training, validation and test data. The test data is set aside and will be used for evaluating performance our model. The train data is the data that we will use for training and validation data is the data that will be used during training to optimize hyper parameters. We performed training on 4 latest YOLO based CNN architectures i.e (YOLOv5 - YOLOv8).

To make it challenging for CNN network and learn from fewer samples, we have randomly picked and split the dataset images into 50-50 i.e half of images will be used for training and rest of the half for testing. The detailed breakup of number of images and classes in each category is given in table 3.1 and 3.2 below for both VAID and AIS dataset.

Table 3.1: Train/Test Split of VAID Dataset

Class	**Train**	**Test**	**Validation**
Sedan	9205	18604	9019
Minibus	111	254	106
Truck	681	1466	715
Pickup	682	1420	657
Bus	127	253	126
Cement Truck	38	88	40
Trailer	200	337	175

Each of the above mentioned network was trained for 100 epochs with a batch size of 4 images, weights initialized to 0.0005 and stochastic gradient descent (SGD) for optimization as shown in table 3.3 below

Table 3.2: Train/Test Split of KIT-AIS Dataset

Class	**Train**	**Test**	**Validation**
Car	1170	1838	616
Truck	4	2	4
Bus	50	39	14
Minibus	36	63	20

Table 3.3: Training Configuration

Sr.No	**Training Parameters**	**Value**
1	Epochs	100
2	Batch size	4
3	Optimizer	SGD
4	Weight Decay	0.0005

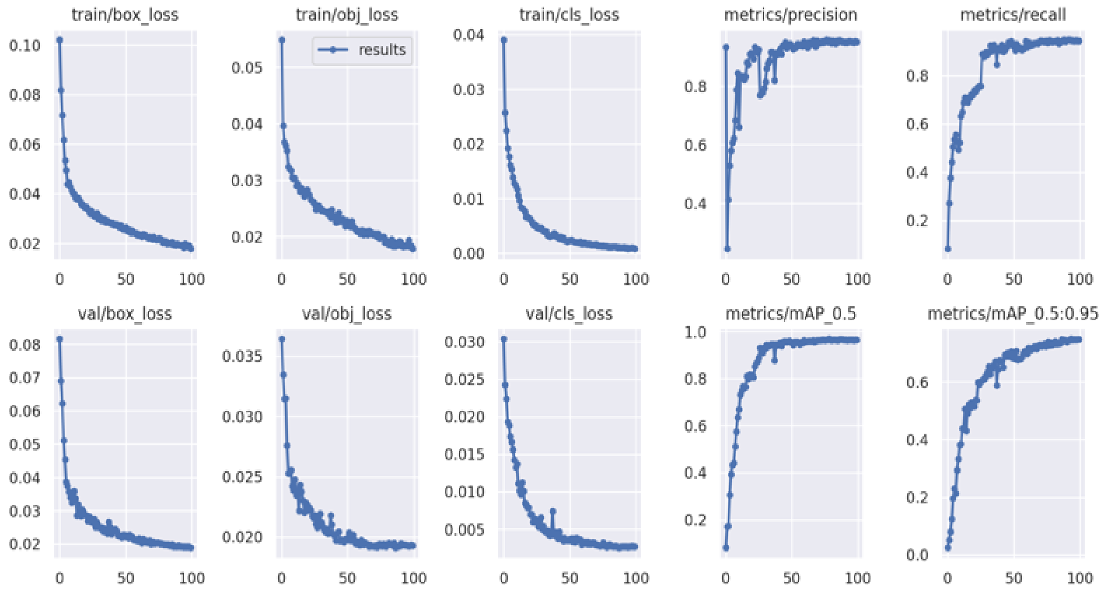


Figure 3.3: Training Results of YOLOv5 on VAID Dataset

The Figure 3.3 and 3.4 shows various graphs that were optimized during training of YOLOv5 network on both VAID and KIT-AIS dataset (i.e The training and validation bounding box estimation loss, the training and validation classification loss, precision and recall metrics and overall mAP).

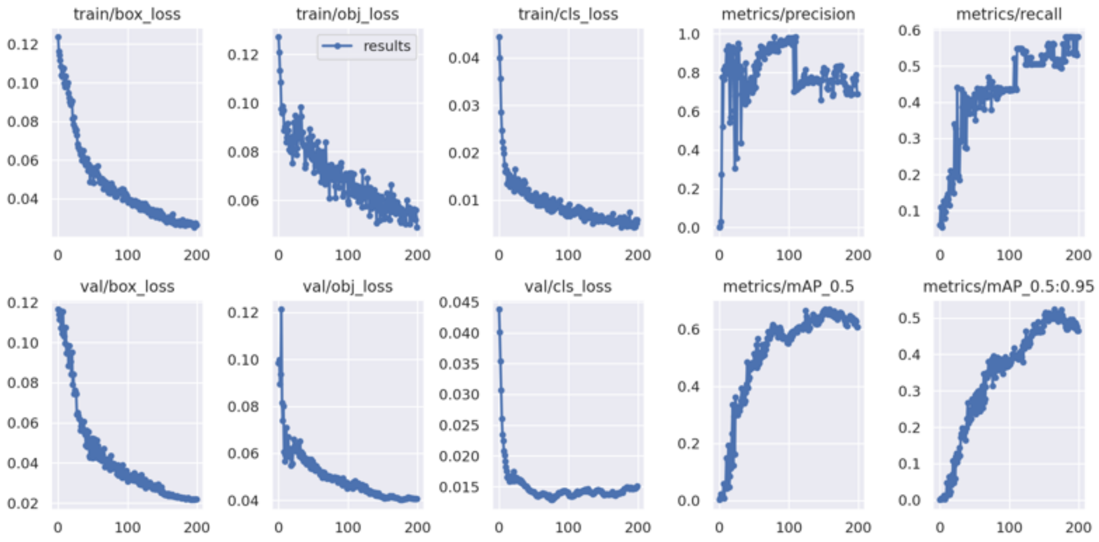


FIGURE 3.4: Training Results of YOLOv5 on KIT-AIS Dataset

3.2 Localization Ensemble

In this chapter the earlier discussed bounding box suppression schemes already being utilized by CNN's to eliminate redundant bounding boxes (i.e non-max suppression, soft NMS etc) are evaluated. The localization ensemble scheme is first discussed, the two of our proposed methods have been implemented for localization ensemble, and the results of each of these is then tabulated. Our two datasets i.e VAID and KIT-AIS dataset are already used to train four YOLO based object detection architectures, i.e YOLOv5 to YOLOv8. Following key points to be discussed in this chapter for both datasets are:

- Evaluate localization performance of individual trained network to set a benchmark for comparison on both datasets.
- Performance of simple NMS technique to benchmark with other bounding box suppression techniques.
- Performance of simple Soft NMS, Softer NMS is also evaluated.

- Our proposed localization ensemble schemes are implemented and evaluated on both datasets.
- Based on comparison we propose the best performing model.

3.2.1 Evaluation Criteria

To evaluate the performance of applied ensembling techniques, evaluation criteria is defined below:

- Average precision (AP), and Mean Average Precision (mAP) are the terms used for comparative analysis.

3.2.2 Case 1: Individual Network Performance

We evaluated each of the trained model independently and so that performance of each network can be compared with the non-max suppression and ensembling schemes to be utilized and set some benchmark scores for these schemes. Table 3.4 and 3.5 below lists down both detection performance metrics of each network on VAID and KIT-AIS dataset.

Table 3.4: Detection Metrics of Each Trained Network on VAID Dataset

Class	**YOLOv5**	**YOLOv6**	**YOLOv7**	**YOLOv8**
Sedan	95.59	45.15	92.36	93.60
Minibus	93.98	36.84	81.30	94.74
Truck	85.58	38.10	77.37	82.38
Pickup	85.78	32.61	68.36	74.85
Bus	93.17	42.73	84.68	91.88
Cement Truck	61.74	3.68	14.68	76.60
Trailer	89.45	38.88	81.06	85.98
mAP	86.47	33.99	71.40	85.72

Table 3.5: Detection Metrics of Each Trained Network on KIT-AIS Dataset

Class	**YOLOv5**	**YOLOv6**	**YOLOv7**	**YOLOv8**
Car	94.70	91.09	96.49	92.53
Truck	3.37	5.50	5.51	9.26
Bus	73.33	71.62	73.19	86.67
Minibus	15.1	8.00	7.42	36.00
mAP	86.47	33.99	71.40	85.72

For the VAID dataset YOLOv5 network is giving better results for most of the categories excluding minibus and cement truck class, for which YOLOv8 is giving best results, as far as KIT-AIS dataset is concerned, it can seen that YOLOv8 is giving better results for most of the classes. The reason for low score of category Truck is due to fact that it had few samples in training phase. So, each network is performing well for a particular class but not all the classes, so if could combine the results of these trained networks it is evident that there will be improvement in the results.

3.3 Non-Maximum Suppression Techniques

Now that we have our bounding box outputs against a test object from each model, There are approaches that allows us to keep the best bounding box and get rid of the redundant bounding boxes. Numerous methods primarily employ non-maximum suppression techniques. The bounding box selection method is a crucial component whose effectiveness will have a direct impact on how accurate the object identification method is. Therefore, the key to successful object detection is a suitable bounding box selection mechanism, Some of the non-maximum suppression techniques include the conventional NMS, soft NMS and its variants. All these algorithms will be applied to results obtained form from each of our model and evaluated afterwards.

3.4 Non-Maximum Suppression

To apply NMS on the test image outputs, obtained after passing test images with each of our trained model for both the datasets, and them perform bounding box scaling to get final output from each vehicle detection network. Then we combine all the outputs for a test image and simply pick the highest score bounding box and suppresses the possible bounding boxes with non-maximum scores based on overlapping threshold (IoU), as described in equation 3.1, after applying this algorithm we evaluate the remaining bounding box outputs and its underlying classes with ground truth annotation of that test image, and repeat the said process for both datasets. Finally we evaluate detection performance metric as per criteria defined for analysis in table 3.6 and table 3.7 for both datasets.

$$\begin{aligned} Pi = \{Pi, iou(Box_k, Box_i) < M_{th} \\ 0, iou(Box_k, Box_i) \geq M_{th}\} \end{aligned} \tag{3.1}$$

Where The bounding box Box_i has class probability score P_i. Box_k has the maximum class probability of any bounding box. The non-maximum bounding box is called Box_i. The overlap threshold is M_{th} taken as 0.5. $iou(Box_k, Box_i)$ is an equation that represents the overlap of two bounding boxes as follows,

$$iou = \frac{Box_k \cap Box_i}{Box_k \cup Box_i} \tag{3.2}$$

Table 3.6: Detection Metrics of NMS on KIT-AIS Dataset

Class	**Average Precision (AP)**
Car	74.47
Truck	5.56
Bus	78.06
Minibus	32.00
Mean Average Precision (mAP)	47.52

Table 3.7: Detection Metrics of NMS on VAID Dataset

Class	**Average Precision (AP)**
Sedan	96.89
Minibus	95.86
Truck	89.19
Pickup	87.58
Bus	93.50
Cement Truck	68.98
Trailer	91.54
Mean Average Precision (mAP)	89.08

3.4.1 Results

The results of conventional NMS technique are on the lower side due to the fact that it is based on single level decision boundary and due to which a lot of boxes get eliminated, and also due to fact that object very close to each might also get deleted causing a miss detection.

3.5 Soft Non-Maximum Suppression

In the soft maximum suppression we reduce the detection score of overlapping candidate boxes gradually by imposing a penalty on them and therefore decreasing the chance to miss detection as mentioned in Eq. 3.3 and Eq. 3.4. The fixed threshold method is difficult to adjust to object detection in complicated environments, as can be observed in the conventional NMS method. As a result, two Soft-NMS approaches based on penalty factors were suggested in the literature review [14]. This approach steadily decreases the number of candidates by lowering the detection score. The following equations show two penalty functions to reduce the detection score,

$$\begin{aligned} Pi &= \{Pi, iou(Box_k, Box_i) < M_{th} \\ Pi &= \{Pi * (1 - iou(Box_k, Box_i)), iou(Box_k, Box_i) \geq M_{th} \end{aligned} \tag{3.3}$$

$$Pi = Pi * e^{\frac{-iou(Box_k, Box_i)^2}{\sigma}} \tag{3.4}$$

where sigma represents the variance of the Gaussian function.

Equation (3.3) is a linear penalty function, and Equation (3.4) is a Gaussian penalty function. The Soft-NMS approach simply lowers the detection score of bounding boxes with high overlap. This technique can, in part, lessen the likelihood of missing detection. The bounding box list will abruptly alter when the overlap degree is close to the threshold value M_{th} because the linear penalty function curve is not continuous. The ideal penalty function should be a continuous smooth curve, where the detection score smoothly declines as the bounding box overlap grows. In light of this, we review a number of recent techniques that have helped the Soft-NMS method's penalty function and results are tabulated in table 3.9,3.10 for VAID and table 3.8 and 3.11 for AIS dataset. The Softer NMS proposed in [14], whose penalty function as follows,

$$Pi = Pi * (1 - iou(Box_k, Box_i))^3 \tag{3.5}$$

In the NMS technique and the modified NMS method, the main consideration in determining whether to keep a bounding box is its detection score exceeds a specific pre-define threshold or whether its overlap IOU value exceeds the predefine threshold.

Table 3.8: Detection Metrics of Soft NMS on KIT-AIS Dataset

Class	**Average Precision (AP)**
Car	97.17
Truck	5.51
Bus	93.22
Minibus	39.64
Mean Average Precision (mAP)	58.88

Table 3.9: Detection Metrics of Soft NMS on VAID Dataset

Class	**Average Precision (AP)**
Sedan	97.20
Minibus	96.37
Truck	90.77
Pickup	88.88
Bus	94.03
CementTruck	78.20
Trailer	92.46
Mean Average Precision (mAP)	91.13

Table 3.10: Detection Metrics of Softer NMS on VAID Dataset

Class	**Average Precision (AP)**
Sedan	97.19
Minibus	96.36
Truck	90.73
Pickup	88.88
Bus	94.02
Cement Truck	78.15
Trailer	92.43
Mean Average Precision (mAP)	91.11

Table 3.11: Detection Metrics of Softer NMS on KIT-AIS Dataset

Class	**Average Precision (AP)**
Car	97.15
Truck	5.51
Bus	93.22
Minibus	39.64
Mean Average Precision (mAP)	58.90

As observed that soft NMS technique performs better than due to the fact that it reduces the detection score gradually to avoid overlapping boxes being drastically deleted and causing a miss detection. It is observed that non linear penalty term introduced as described in Eq. 3.5, definitely improves performance, because now we are not directly eliminating bounding boxes but picking the highest detection box gradually.

3.6 Localization Ensemble 1

3.6.1 Overlapping Bounding Box Ensemble

This is one of the proposed scheme in which the candidate box position is adjusted based on weights computed from non maximum boxes, with intent to increase the detection accuracy of the bounding box in Eq 3.8 by computing weights based on IOU overlap ratio as mentioned in Eq. 3.7. Each candidate box vector ($Box = [x1, y1, x2, y2, s, c]$) provides six position details about the object, including the object category c, detection score s, and the top left and lower right corners of the candidate box. Each parameter has a direct impact on the method's overall detection performance. To limit the amount of calculation in this method, candidate boxes with detection scores below a defined threshold Mt are first deleted. The equation is as follows:

$$\begin{gathered} Pi = \{Pi, Pi < M_{th} \\ 0, Pi < M_{th}\} \end{gathered} \tag{3.6}$$

The bounding box set $C = \{C1, C2, .., Ci, .., Cn\}$ that has a significant amount of overlap with the best bounding box is regarded as detecting the same object. The basic box in this set of bounding boxes is determined by this method using the bounding box with the highest detection score, $Box_k = [x1, y1, x2, y2]$. This bounding box fusion approach gives each bounding box a varied weight since the likelihood that it contains object information varies. A possible bounding box is given a larger fusion weight w_i because it is thought that the more overlap there is between the bounding boxes and best box Box_k, the closer it is to the actual object box. function is defined as follows,

$$W_i = \frac{iou(Box_k, Box_i)}{\sum_{n=1}^{n} iou(Box_k, Box_i)} \tag{3.7}$$

Each possible bounding box's weight is calculated by dividing the IOU value between it and best bounding box Box_k by the sum of the IOU values between all candidate boxes and best box Box_k. We can create a new candidate box position information by weighting and summing the bounding boxes after learning the weight of each bounding box. The following is the computation process:

$$Box_k = Box_k - \sum_{n=1}^{n} \frac{(Box_k - C_i) * w_i}{\sum_{n=1}^{n} w_i} \tag{3.8}$$

The new bounding box's class probability and class c match those of the best bounding box Box_k. The best bounding box's position can be adjusted by the Overlapping bounding box ensemble method.

Table 3.12: Detection Metrics of Overlapping Bounding Box Ensemble on VAID Dataset

Class	Average Precision (AP)
Sedan	96.93
Minibus	95.86
Truck	89.26
Pickup	87.42
Bus	93.88
Cement Truck	68.98
Trailer	91.82
Mean Average Precision (mAP)	89.30

Table 3.13: Detection Metrics of Overlapping Bounding Box Ensemble on KIT-AIS Dataset

Class	Average Precision (AP)
Car	97.31
Truck	7.86
Buss	93.33
Minibus	28.00
Mean Average Precision (mAP)	56.63

3.6.2 Results

As observed that there is some improvement in performance metric due to fact that now the algorithm has adjusted the position of candidate boxes and made it more closer towards ground truth, also it has removed the false positive or non maximum boxes as well.

3.7 Localization Ensemble 2

3.7.1 Class Score Bounding Box Ensemble

This is the second proposed scheme in which the bounding box position is adjusted based on weights computed from possible bounding boxes, with intent to increase the detection accuracy of the said bounding box, Class score based ensemble algorithm is similar to overlapping bounding box ensemble, in a way it can shrink or expand the bounding box in the direction of each possible bounding box as described in Eq 3.10 but the difference here is that it computes weights based on class probability as mentioned in Eq. 3.9. The Class score based ensemble method also chooses a new best box from the output of available bounding boxes; the detection class probability s denotes the likelihood that the new bounding box contains a genuine object. The bounding box with the most information is the only one kept using the NMS approach, and the bounding box with the least information is deleted. The final detection effect will be significantly improved if we can incorporate the bounding box with little information into the maximum bounding box. he Class score based ensemble method continues to incorporate the location data of the best bounding box into the possible bounding box. The possible bounding box set $C = C1, C2, .., Ci, .., Cn$ that significantly overlaps the best bounding box is regarded by the Class score based ensemble algorithm as detecting the same item. The best bounding box $Box_k = [x1, y1, x2, y2]$ is regarded by the this technique as the fundamental box in the set of possible bounding boxes. The

Class score based ensemble approach gives each possible bounding box a variable weight during the bounding box frame fusion procedure because the probability that the bounding box includes object information varies. The higher the class probability s, the closer it is thought to be to the genuine object, and as a result, it is given a higher fusion weight w_i.

$$w_i = \frac{Pi}{\sum_{n=1}^{n} Pi} \tag{3.9}$$

Each possible bounding box's weight is calculated by dividing its individual class probability by the total of all possible bounding boxes class scores. We may create a new final bounding box position information by weighing and summing the possible bounding boxes after learning their weights. The calculation procedure is as follows:

$$Box_k = Box_k - \sum_{n=1}^{n} \frac{(Box_k - C_i) * w_i}{\sum_{n=1}^{n} w_i} \tag{3.10}$$

This final bounding box's class score s and class c match those of the first best bounding box Box_k respective class score and category. The adjustment technique of the Class score based ensemble method is comparable to the overlapping bounding box ensemble approach for the object information that may present in each bounding box. The position of the best bounding box is compressed or stretched in every direction by the algorithm. Finally, a new bounding box that completely envelops the real object is obtained.

Table 3.14: Detection Metrics of Class Score Based Ensemble on KIT-AIS Dataset

Class	**Average Precision (AP)**
Car	97.30
Truck	7.86
Buss	93.26
Minibus	27.46
Mean Average Precision (mAP)	56.50

Table 3.15: Detection Metrics of Class Score Based Ensemble on VAID Dataset

Class	Average Precision (AP)
Sedan	96.94
Minibus	95.32
Truck	89.26
Pickup	87.23
Bus	91.48
Cement Truck	68.67
Trailer	91.32
Mean Average Precision (mAP)	88.60

3.7.2 Results

It is pertinent to use the proposed schemes (i.e. Overlapping bounding box ensemble and Class score based ensemble) due to the reason that these proposed schemes are now not only adjusting the position of bounding boxes to make them more accurate.

It can be observed that among the YOLO based networks that we have trained on VAID dataset, it was observed that YOLOv5 gave the best detection score for most of the classes except for minibus and cement truck classes, for which YOLOv8 was giving better results, so we take the results of YOLOv5 and benchmark with both of our localization ensemble schemes on VAID dataset, on the other hand YOLOv8 give best individual network results, so its results are used to benchmark with our proposed localization ensemble schemes on KIT-AIS dataset.

TABLE 3.16: Detection Metrics of YOLOv5 w.r.t our Ensemble Schemes on VAID Dataset

Class	**YOLOv5**	**OBBE**	**CSBE**
Sedan	95.59	96.93	96.94
minibus	93.98	95.86	95.32
truck	85.58	89.26	89.26
pickup	85.78	87.42	87.23
bus	93.17	93.88	91.48
cement truck	61.74	68.98	68.67
trailer	89.45	91.82	91.32
Accuracy	86.47	89.30	88.60

TABLE 3.17: Detection Metrics of YOLOv8 w.r.t our Ensemble Schemes on KIT-AIS Dataset

Class	**YOLOv5**	**OBBE**	**CSBE**
Car	92.53	97.31	95.54
Truck	9.26	7.86	7.86
Bus	86.67	93.33	93.26
Minibus	36.00	28.00	27.46
Accuracy	56.12	56.63	56.50

3.7.3 Results

It can be seen that there is improvement in detection results by applying the proposed scheme and our localization ensemble increase mean average precision (mAP) by 2.83% for VAID and 0.51% for KIT-AIS. Therefore we can say that our proposed schemes can be utilized in scenarios, where a single objection detection network is not enough to meet the system requirements and so then multiple detection CNNs can be trained for a particular problem and their results can be combined using our proposed schemes.

Both localization ensemble algorithms place a strong emphasis on retaining local maximum candidate boxes, eliminating redundant boxes, and adjusting the position of the maximum candidate box by reusing redundant boxes. By increasing the amount of real object information in the maximum candidate box, the detection accuracy can be improved by increasing the overlap between the candidate box and the real box.

3.8 Chapter Summary

In this chapter, our proposed techniques is tested under different scenarios, which include conventional NMS, Soft NMS, Softer NMS, overlapping bounding box ensemble and class score based bounding box ensemble methods. The comparison of proposed techniques is made between the known studies on both the datasets, it was observed that our proposed techniques perform fairly well compared to that of a single CNN.

Chapter 4

Classification Ensemble

In this section, our two classification ensembling techniques will were discussed, implemented, then the results are compiled and tabulated. Our two datasets i.e VAID and KIT-AIS dataset is already used to train five YOLO based object detection architectures, i.e YOLOv5 to YOLOv8.

Following key points to be discussed in this chapter for both datasets are:

- Evaluate classification performance of individual trained network to set a benchmark for comparison on both datasets.
- Performance of both classification Ensemble techniques is evaluated on both datasets.
- Based on comparison we propose the best performing model.

4.1 Evaluation Criteria

To evaluate the performance of applied ensembling techniques, evaluation criteria is defined below:

- Accuracy, precision, recall and F1 score are the terms used for comparative analysis.

4.2 Case 1: Individual Network Performance

We evaluate classification performance of each of the trained model independently, so that performance of each network can be compared with the proposed ensembling schemes, to be utilized to set some benchmark scores for these schemes. Table 4.1 lists classification performance metrics of each network on VAID dataset, and table 4.2 lists down both detection and classification performance metrics of each network on KIT-AIS dataset.

Table 4.1: Classification Metrics of Each Trained Network on VAID Dataset

Class	**YOLOv5**	**YOLOv6**	**YOLOv7**	**YOLOv8**
Sedan	98.27	96.10	94.67	96.69
minibus	99.75	99.44	98.95	99.72
truck	98.15	96.00	95.50	97.14
pickup	97.98	95.43	91.98	96.62
bus	99.73	99.23	98.49	99.64
cement truck	99.31	98.94	97.57	99.11
trailer	99.35	98.57	98.03	99.00
Accuracy	98.93	97.67	96.46	98.28
Precision	0.846	0.647	0.583	0.789
Recall	0.846	0.647	0.583	0.789
F1 Score	0.846	0.647	0.583	0.789

Table 4.2: Classification Metrics of Each Trained Network on KIT-AIS Dataset

Class	**YOLOv5**	**YOLOv6**	**YOLOv7**	**YOLOv8**
Car	96.57	97.47	96.98	97.98
Truck	99.60	99.60	99.60	99.60
Bus	99.49	98.99	99.70	99.60
Minibus	97.47	97.68	97.48	98.38
Accuracy	98.28	98.43	98.44	98.89
Precision	0.491	0.696	0.483	0.728
Recall	0.455	0.486	0.482	0.573
F1 Score	0.471	0.493	0.483	0.613

We have evaluated classification performance metrics by first finding the corresponding outputs for a object in a test image by IOU of 0.5, and checking the class prediction with ground truth class, if there was single output then it is taken as final class, and there were multiple redundant outputs then each of these prediction is considered with respect to ground truth, therefore there is one correct prediction and rest are mere false positives.

4.3 Classification Ensemble 1: Class Score Weighted Averaging Scheme (SWA)

The first step in performing classification ensemble is to extract and group together similar predictions against a target object in test image among all the trained networks, therefore again IOU metric is used to show how much overlap exists between two bounding boxes, so now we take all these predictions and their class scores, and combine the class predictions by aggregating class score for each category and then selecting the class with highest aggregate score as final prediction, hence called class score weighted averaging scheme as shown in figure 4.1

below. The result of classification ensemble discussed above is shown in table 4.3 and table 4.4 for VAID and KIT-AIS datasets.

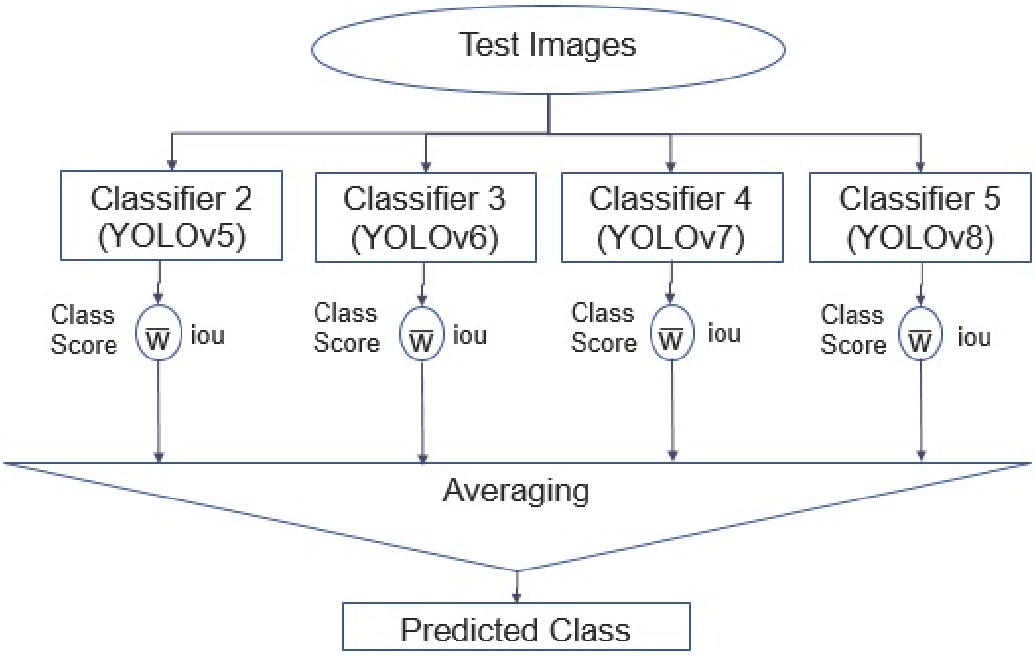


FIGURE 4.1: Block Diagram Classification Ensemble 1

TABLE 4.3: Ensemble 1 Classification Metrics on VAID Dataset

Class	**Ensemble 1**
Sedan	98.64
minibus	99.74
truck	99.31
pickup	99.16
bus	99.72
cement truck	99.77
trailer	99.74
Accuracy	99.49
Precision	0.870
Recall	0.870
F1 Score	0.870

Table 4.4: Ensemble 1 Classification Metrics on KIT-AIS Dataset

Class	**Ensemble 1**
Car	97.69
Truck	99.60
Bus	99.70
Minibus	98.19
Accuracy	98.80
Precision	0.735
Recall	0.553
F1 Score	0.594

4.4 Classification Ensemble 2: Class Score and IOU Weighted Averaging Scheme (SIWA)

The difference between this method and previous is that here we have taken both class score and IOU ratio before aggregating, so now the sum is made with class score and IOU ratio multiplied for each class and applying class wise aggregate hence called class score and IOU weighted averaging scheme as shown in figure 4.2.The logic behind using IOU is that IOU will impose a penalty term on boxes detected that are away and give more weightage to boxes that are close to test object. Lets look at an example, suppose we get three bounding boxes with their respective class and class score for a particular object in a test image, the class scores are [0.90 0.80 and 0.10] and respective class label are [2 1 5], we compute the IOU between the test object and these bounding boxes to be [0.60 0.90 0.50], so now we compute the weights by simply multiplying IOU metrics with class scores to get [0.54 0.72 0.05], we pick the class corresponding to the highest score i.e 0.72 and therefore class 1 is selected as the final class which is sedan class in our case, after score and IOU weighted average scheme. The result of our second classification ensemble technique is given in table 4.5 for VAID and 4.6 for KIT-AIS dataset below.

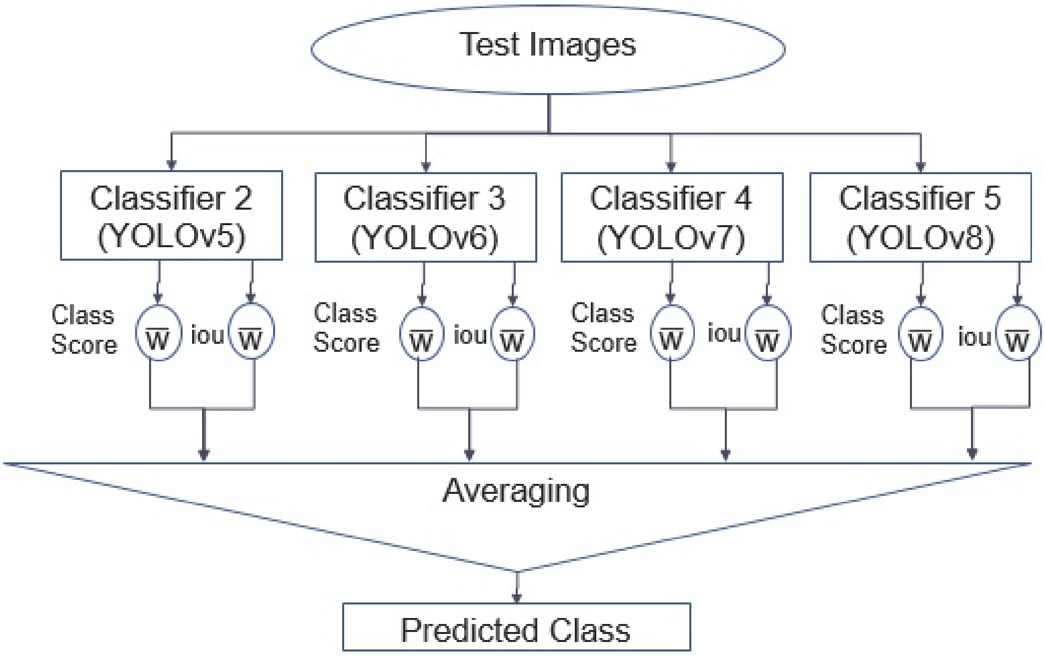


FIGURE 4.2: Block Diagram Classification Ensemble 2

TABLE 4.5: Ensemble 2 Classification Metrics on VAID Dataset

Class	**Ensemble 2**
Sedan	98.68
minibus	99.78
truck	99.33
pickup	99.23
bus	99.78
cement truck	99.71
trailer	99.71
Accuracy	99.46
Precision	0.866
Recall	0.866
F1 Score	0.866

TABLE 4.6: Ensemble 2 Classification Metrics on KIT-AIS Dataset

Class	**Ensemble 2**
Car	97.69
Truck	99.60
Bus	99.70
Minibus	98.19
Accuracy	98.79
Precision	0.735
Recall	0.8553
F1 Score	0.594

We can see there is decent improvement in classification metrics by using both of our proposed ensemble classification methods, due to the reason that now not only that we are eliminating false positives, but also using the information obtained from redundant boxes and there corresponding class predictions in correcting final class prediction.

4.5 Chapter Summary

In this chapter, our proposed classification ensemble techniques is tested under different scenarios. It was observed that our proposed ensemble techniques fairly improves upon the classification performance of a single CNN.

Chapter 5

Results and Discussion

5.1 Combined Ensemble Scheme

After analysis of results obtained in chapter 3 and 4 by individually applying localization and classification ensemble schemes, we propose a combined ensemble model which utilizes best performing algorithm in both the schemes to get final output as shown in figure 5.2. This combined ensemble process is essentially taking outputs from the YOLO based trained networks grouping together similar detections i.e bounding boxes based on IOU metric and then passing that group to two separate blocks simultaneously i.e Ensemble for detection and Ensemble for Classification. The Classification Ensemble block process the class scores and class label information to find the final class prediction, on the other hand detection ensemble block estimates the final bounding box, the results from both blocks are then combined to output the final bounding box and class prediction, also the pre-processing step in which image and bounding box scaling is done to match the original image resolution and ground truth annotation is shown figure 5.2 below.

As far as detection is concerned we have chosen overlapping bounding box fusion based ensemble model which is based on IOU measure between best bounding box and other associated bounding boxes to compute new position of this best bounding box, as it was found to be giving better results compared to other schemes, for classification we have selected class score based aggregation ensemble scheme which simply sums the class wise scores from each CNN network for a particular test object, and picks the class with highest score after aggregating.

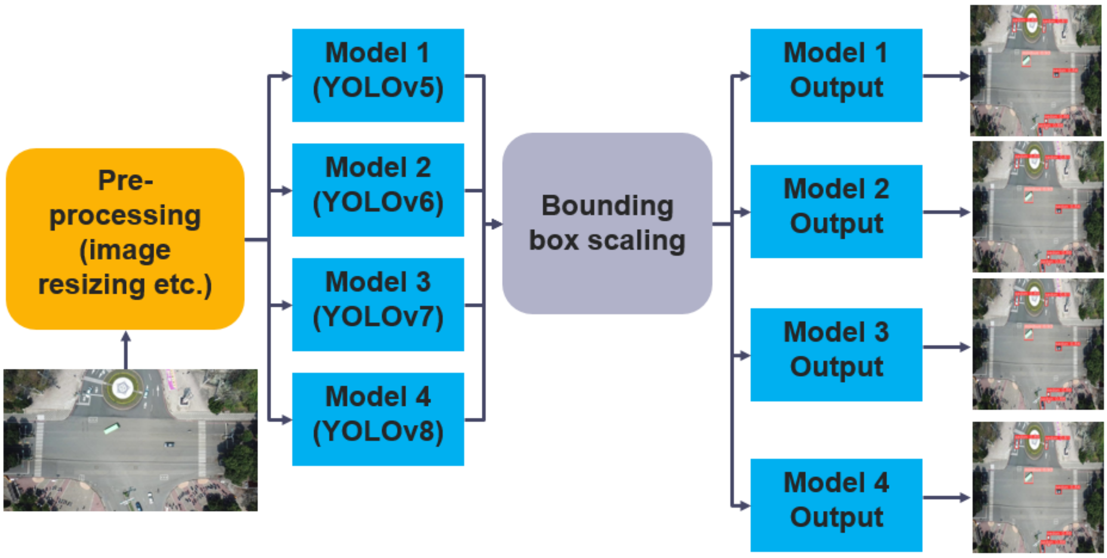


FIGURE 5.1: Preprocessing Step for Ensemble

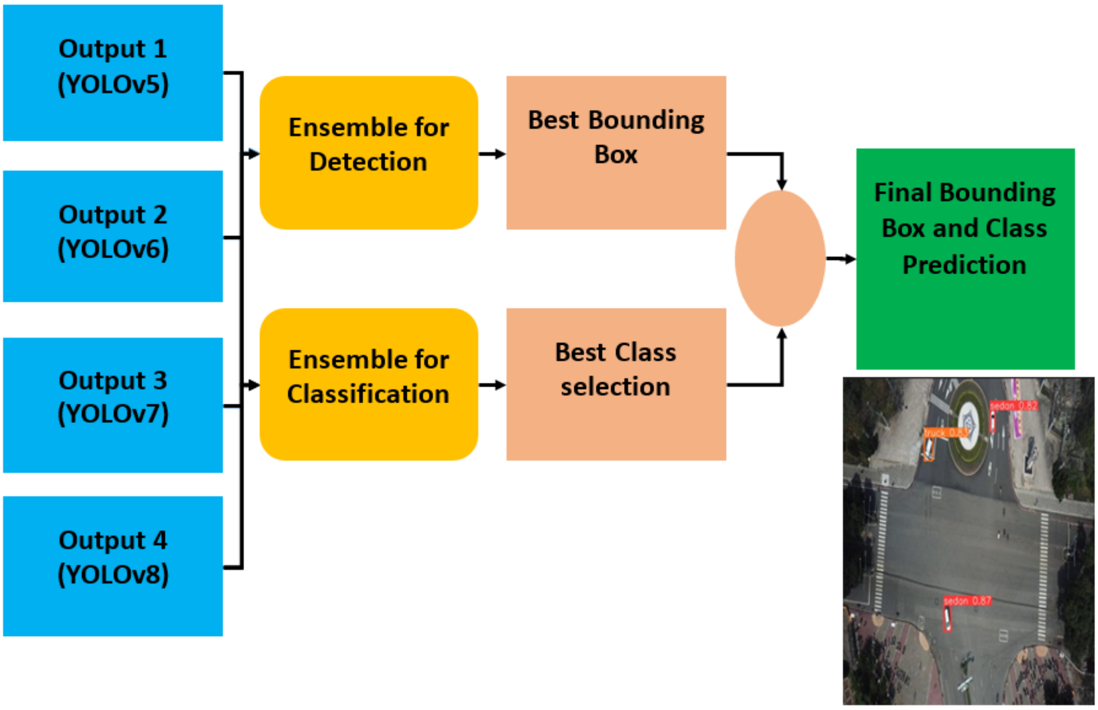


FIGURE 5.2: Combined Ensemble Model Block Diagram

5.2 Detection Performance

Table 5.1 and 5.2 below lists down localization performance metrics on VAID and KIT-AIS dataset using our combined ensemble model below:

Table 5.1: Detection Metrics on VAID Dataset

Class	**Average Precision (AP)**
Sedan	96.95
Minibus	91.92
Truck	89.25
Pickup	86.75
Bus	93.77
Cement Truck	69.98
Trailer	91.81
Mean Average Precision (mAP)	88.50

Table 5.2: Detection Metrics on KIT-AIS Dataset

Class	**Average Precision (AP)**
Car	97.31
Truck	7.86
Bus	93.33
Minibus	28.00
Mean Average Precision (mAP)	56.63

It can be observed that our proposed method is a decent improvement compared to single network result, the reason being that the algorithm is not only retaining the best bounding box but also using the information of other redundant boxes to adjust the position of this best bounding box, and after that eliminating these redundant boxes, which if had been left present, would have resulted in lower detection precision because of them being false positives. The best results are obtained for sedan class in VAID dataset due to the fact it had highest numbers of samples in training data, and cement truck had the minimum score due to smaller number of samples, same is the case in KIT-AIS dataset where car class has the highest score and Truck class has the lowest score due to smallest number of samples.

5.3 Classification Performance

Table 5.3 and 5.4 lists down classification performance metrics on VAID and KIT-AIS dataset using our classification ensemble method below:

Table 5.3: Classification Metrics on VAID Dataset

Class	**Accuracy**
Sedan	98.64
minibus	99.74
truck	99.31
pickup	99.16
bus	99.72
cement truck	99.77
trailer	99.74
Accuracy	99.49
Precision	0.870
Recall	0.870
F1 Score	0.870

TABLE 5.4: Classification Metrics on KIT-AIS Dataset

Class	**Accuracy**
Car	97.69
Truck	99.60
Bus	99.70
Minibus	98.19
Accuracy	98.80
Precision	0.735
Recall	0.578
F1 Score	0.610

It can be observed that our classification ensemble method is giving significant improvement compared to single network result, main reason being that the algorithm utilizing best performing network by aggregating the class predictions of redundant boxes against a given object, before eliminating those redundant boxes, so now the true class having lower class confidence among the other class predictions with higher scores due to lack of network optimization and might not be taken as final class, still can be taken as final class prediction by averaging the class predictions among all networks resulting in improved performance.

5.4 Comparison with Known Studies

The publishers of VAID dataset in 2020 had trained and evaluated few CNN based object detection networks on their dataset [1], and this is the only known study made in this context of vehicle detection in aerial images, and they found YOLOv4 to give the best detection score, including some other networks as well which perform fairly well, so we take those results and benchmark with our ensemble model on the VAID dataset, as we have also used the same train and test split that was used by authors of VAID, and shown in table 5.5 below.

Table 5.5: Detection Metrics of YOLOv4 w.r.t Proposed Ensemble Scheme on VAID

Class	YOLOv4	U-Net	MobileNetv3	RefineDet	Proposed Ensemble Model
Sedan	98.49	67.20	70.46	89.08	96.95
minibus	96.04	94.36	89.02	90.14	91.92
truck	96.44	83.46	64.92	82.21	89.25
pickup	57.25	82.20	75.73	84.59	86.75
bus	97.03	97.84	87.67	90.46	93.77
cement truck	69.94	91.24	90.30	80.68	69.98
trailer	95.45	80.74	78.14	86.64	91.81
mAP	87.23	85.38	79.46	86.26	88.50
Precision	0.90	0.909	0.281	0.242	0.870
Recall	0.92	0.901	0.880	0.964	0.930
F1-Score	0.91	0.905	0.417	0.373	0.892

5.4.1 Discussion

It can be seen that there is considerable improvement in both detection and classification metrics of our proposed ensemble scheme, therefore we can say that our proposed schemes can be utilized in scenarios, where a single objection detection network is not enough to meet the system requirements, and then multiple detection CNNs can be trained for a particular problem and their outputs can be combined in a post processing step using our proposed method. The advantage of our proposed scheme is that due to the fact that it is applied in post processing step, the said algorithm is limited to a particular CNN network architecture, so any CNN network can be utilized for the ensemble modeling which are trained separately, there is no need to retrain the model.

5.5 Chapter Summary

Following key points can be summarized from this chapter:

- A two-stage hierarchical ensemble model approach was then proposed in which the best bounding box was computed using the best bounding box and its overlapping redundant boxes in the localization ensemble phase, and similarly classification ensemble is performed on class predictions associated each of the overlapping redundant boxes to get final class prediction.
- Performance comparison of above mentioned techniques with known studies
- Detection average precision of our proposed scheme increased by 2.83% and Classification accuracy of our proposed classification scheme increased by 0.56% for VAID dataset.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this chapter, a complete dissertation is summarized and future research areas are defined. The aim is to facilitate the research directions of interesting readers in the field of vehicle detection in aerial images.

In this study two-stage ensemble based approach is proposed to localize and classify vehicles in aerial images. Firstly multiple YOLO based object detection networks were trained on VAID and KIT-AIS datasets. The training of different architectures involves pre-processing of dataset images (i.e., resizing, scaling etc.) and also conversion of its related annotations in different formats. The results for localization and classification were then thoroughly evaluated. Ensemble techniques for localization and classification were performed separately, using overlapping bounding box fusion for localization, and class score weighted averaging scheme for classification. A two-stage hierarchical ensemble model approach was then proposed in which the best bounding box was computed using the best bounding box and its overlapping redundant boxes in the localization ensemble phase, and similarly classification ensemble is performed on class predictions associated each of the overlapping redundant boxes to get final class prediction. The results of the proposed scheme were then evaluated against the individual network and it was shown that the proposed system has better performance for both detection and classification.

Detection average precision of our proposed scheme increased by 2.83% for VAID dataset and 0.51% for KIT-AIS, and on the other hand classification accuracy of our proposed classification scheme increased by 0.56% for VAID dataset, which is a notable improvement in this field.

6.2 Future Work

The images used in the datasets were mostly taken in clear daylight, good weather conditions and at a lower altitude as well (i.e., 100 to 300 meters), the images were not always parallel to image axis and sometimes rotated, these are the few constraints we defined at the beginning of our work, Therefore the proposed scheme can be implemented and tested for diverse weather conditions and higher altitude images like satellite images to see its robustness. The proposed architecture can be implemented on off the shelf AI development kits (i.e NVIDIA jet-son etc.) to be utilized for real time AI applications. NVIDIA jetson nano is a low cost and small embedded computer as shown in figure 6.1 that is used to run AI software.

It comes in a development board with other peripherals like on-board camera, storage, Ethernet, display and USB support. It comes with Quad-core ARM Cortex-A57 processor and has NVIDIA GPU support to run neural network architectures, to train and deploy AI based applications. Orientation information can also be incorporated by computing the principal orientation first and then image is rotated in such a way that it becomes parallel with image axis before applying our proposed algorithm, with the intent that the rectangular bounding box perfectly captures the object. Another approach could be to optimize different light weight CNN architectures like efficientNet, mobileNet etc., and apply the proposed ensemble technique to validate the results.

Figure 6.1: NVIDIA Jetson Nano Development Board

Bibliography

[1] C.-Y. Li, K.-C. Tu, and H.-Y. Lin, "Vaid an aerial image dataset for vehicle detection and classification," *IEEE Access*, vol. 8, pp. 212 209 – 212 219, 2020.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893 vol. 1, 2005.

[3] Y.-T. W. Chung-Hsien Huang, J.-H. Kao, M.-Y. Shih, and C.-C. Chou, "A hybrid moving object detection method for aerial images," *Advances in Multimedia Information Processing*, vol. 6297, p. 357–368, 2010.

[4] S. Razakarivony and F. Jurie, "Vehicle detection in aerial imagery: A small target detection benchmark," *Visual Communication and Image Representation*, vol. 34, pp. 187–203, 2016.

[5] A. W. S. T. Nathan Mundhenk, G. Konjevod and K. Boakye, "A large contextual dataset for classification detection and counting of cars with deep learning," *Computer Vision–ECCV*, vol. 1, pp. 785–800, 2016.

[6] J. D. Z. Z. S. B. J. L. G.-S. Xia, X. Bai, "Dota: A large-scale dataset for object detection in aerial images," *Conference Computer Vision Pattern Recognition*, vol. 1, pp. 3974–3983, 2018.

[7] Shao, Yang, and L. Liu, "Car detection from high-resolution aerial imagery using multiple features," *International Geoscience and Remote Sensing Symposium*, vol. 1, pp. 4379–4382, 2012.

[8] R. C. H. Yalcin, M. Hebert and M. J. Black, “A flow-based approach to vehicle detection and background mosaicking in airborne video,” *International Conference on Computer Vision and Pattern Recognition*, vol. 1, p. 1202, 2005.

[9] Ross and Girshick, “Fast r-cnn,” *International Conference on Computer Vision (ICCV)*, vol. 1, p. 1440–1448, 2015.

[10] Ren, Shaoqing, He, Kaiming, Girshick, Ross, and S. Jian, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *28th International Conference on Advances in Neural Information Processing Systems*, vol. 1, pp. 91–99, 2015.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 779–788, 2016.

[12] Sommer, Schuchert, and Beyerer, “Fast deep vehicle detection in aerial images,” *Conference on Applications of Computer Vision*, vol. 1, pp. 311–319, 2017.

[13] Benjdira, T. Khursheed, A. Koubaa, A. Ammar, and K. Ouni, “Car detection using unmanned aerial vehicles: Comparison between faster r-cnn and yolov3n,” *1st International Conference on Unmanned Vehicle Systems-Oman (UVS)*, vol. abs/1812.10968, 2019.

[14] J. Cao, W. Ren, H. Zhang, and Z. Chen, “Candidate box fusion based approach to adjust position of the candidate box for object detection,” *IET Image processing*, vol. 15, pp. 2799–2809, 2021.