Automated Overtaking Assistance System: A Real-Time Approach Using Deep Learning Techniques

by

Musarrat Moonjarin
19101586
Krity Haque Charu
19101173
Kh. Fardin Zubair Nafis
19301007
Suraya Jahan Sawly
19101383

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
BRAC University
March 2023

© 2023. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Moonjarin	Krity Haque Charu
Musarrat Moonjarin 19101586	Krity Haque Charu 19101173
Kh. Fardin Zubair Nafis	Sawly
Kh. Fardin Zubair Nafis 19301007	Suraya Jahan Sawly 19101383

Approval

The thesis titled “Automated Overtaking Assistance System: A Real-Time Approach Using Deep Learning Techniques” submitted by

1. Musarrat Moonjarin(19101586)
2. Krity Haque Charu(19101173)
3. Kh. Fardin Zubair Nafis(19301007)
4. Suraya Jahan Sawly(19101383)

Of Spring, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on March, 2023.

Examining Committee:

Supervisor:
(Member)

Md. Khalilur Rhaman

Md. Khalilur Rhaman,PhD
Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Tanzim

Md Tanzim Reza
Lecturer
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

We pledge to uphold the highest ethical standards while performing the research for this thesis by getting the informed consent of every participant, protecting the privacy of sensitive data, and acting honestly and impartially at all times.

Abstract

Road accidents are one of the major causes of fateful deaths in Bangladesh. In most cases it is caused by Overtaking on highways or on regular roads. In terms of overtaking the major task is to decide whether the overtaking is safe or not. There has been a lot of work considering autonomous communication in this field. However, the challenges in terms of Bangladesh, need different and user-friendly solutions. Considering the interest of researchers in this area and to introduce a different evolutionary trend in Bangladesh we approached this research. In this research our basic concept is to suggest the safe overtaking decision to the host drivers considering an overall idea of the environment. Our work aims to decrease early and unfortunate deaths caused by vehicle's abrupt overtaking on Bangladesh's Highways by assisting drivers based on deep learning techniques. Furthermore, the Autonomous system considers communication between vehicles to decide safe overtaking within minimum time.V Vehicle detection and classificationdetection and classification is done using the YOLO model. After measuring the distance and relative velocity, our model suggests the decision by using the help of other significant models used in our research. For distance measurement, we used SegNet and a distance measurement model. In addition, for getting relative velocity we have used optical flow and also for checking whether the driver is on the right lane or not, we have used the PiNet model for lane detection. Moreover, we have no use of other sensors besides the camera and kept only one camera in our proposed system. So in future, users will get this autonomous system in their vehicles at low cost as our system proposes. Experimental results from the proposed system show that deep learning process is better in terms of our country.

Keywords: Image Processing, Overtaking, Autonomous, Distance Measure, SegNet Model, PINet, Optical Flow, YOLO, Deep Learning

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, to our supervisor Khalilur Rahman and co-supervisor Tanzim Reza sir for their kind support and guideline.

And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ϵ	Epsilon
$ADAS$	Advanced Driving Assistance System
$BRTA$	Bangladesh Road Transport Authority
$NHTSA$	National Highway Traffic Safety Administration
RSF	Road Safety Foundation

Chapter 1

Introduction

Bangladesh has seen 413 deaths along with 532 injuries only because of road accidents in November last alone [29]. The country placed 106th for having the most road accident-related deaths among 183 countries [24]. The Road Safety Foundation (RSF) indicated ten major causes behind the growing number of road accidents which includes reckless driving, incompetence and illnesses of drivers, movement of low-speed vehicles on highways, reckless motorcycling by youths, the tendency of flouting traffic rules etc. [29]. Bangladesh Road Transport Authority (BRTA) claimed overtaking, overload and over speed are the major causes of road accidents [7]. When it comes to overtaking, the most difficult task is to decide whether the overtaking is safe or not. Abrupt decisions cannot assure safety. So, the main inspiration for this thesis came from preventing the unfortunate deaths caused in highways because of overtaking and over-speeding issues. Although there exists a lot of work for autonomous vehicles, Bangladesh has seen less researches regarding automated vehicles assistance. Thus, there are many challenges to address for driver assisted vehicles.Our technology uses some deep learning methods to determine the vehicle's distance, speed, and lane ahead. To determine whether it is safe to overtake or not, the information gathered of these is examined. The deep learning methods we have used includes PiNet, SegNet with Monocular Estimation, and Optical Flow. In the chapters that follow, we'll go into further details on each of these elements.

1.1 Research Problem

Advanced Driving Assistance System (ADAS) is mostly used for autonomous vehicles. ADAS has many advantages yet the cost and limited availability beyond higher-end models, remains as the main limitation. It is critical to offer the driver a perspective of the surrounding vehicles in order to provide additional help. Besides, using ADAS to maintain the traffic rules and control road accidents can not assure safe overtaking, knowledge about a vehicle's distance, position, velocity and also the gap between two vehicles are also needed [6].

Articles like [3], [26] implements where the vehicle driver gets assisted for overtaking in the night time through sensors. However, in our country the rainy season stays more than half a year and implementing these kinds of costly sensors in developing countries like Bangladesh will not help us to reduce this problem [3].
Moreover, to give commands about a safe overtaking system, we have to detect real

time objects. We could detect any object by the Raspberry Pi camera module using an advanced classifier[22]. However, several challenges must be addressed when developing an object detection system using IoT and machine learning, including optimizing the algorithms for real-time performance, handling noisy sensor data, and ensuring privacy and security.
According to [18] paper, it is mentioned that using Python, openCV, triangulation method and Flask, we can predict whether an ego-vehicle can overtake or not. This paper can be a good solution for our problem but this technology will fail to detect if there are more than one vehicle from the ego-vehicle and the distance between those vehicles.
Furthermore,[12],[26] share knowledge about vehicle to vehicle (V2V) communications methods. However, if we use this technology then we have to set this technology in every car which is costly and also it can track where we are going which can also be a big issue for privacy. [26], they have made an overtaking system for night but their constraints are: vehicle cannot detect which are in parking mode and has limitation in detecting different vehicles.

According to [27] paper, overtaking decisions are made by using image processing and deep learning models. Here they have used CNN for object detection which can be less accurate in giving results and the model they have used for image processing may not properly work in real time also.
In [14] paper, the system was built with the help of 5G communication with an Adhoc network for communication and micro-controller for giving information about overtaking. Though this can help us in giving faster communication but in our rural area because of network conditions this kind of techniques will not be helpful. Also it has some cost issues. Besides, reading [20] paper, though motion cue obtained from optical flow can be used for overtaking command but this paper has some limitations like detecting road lanes, giving less performance when the vehicles make turns etc. whereas from [15] we can better performance for using RCNN than Raspberry Pi camera module for object detection. Afterwards, from [38] paper, though we are familiar with new methodology using optimal trajectory schemes, this methodology is only for automated systems.

As we are living in a country like Bangladesh where traffic signals and road rules are not well maintained by drivers, a system like “giving command of safe overtaking” can be a vital solution for this problem. This solution can help us to reduce the vast number of road accidents and unwanted life-killing incidents on the road. The shortage of published papers and the lack of proper methods to implement this system are making us well behind to get a perfect solution in our country.

1.2 Research Objectives

This research aims to decrease premature and unfortunate deaths caused by vehicle’s abrupt overtaking on Bangladesh’s Highways by assisting drivers based on implying the deep learning methods. By taking into account variables like the speed and separation between the vehicles, the state of the road, and any potential barriers or hazards that may be present, such a model can assist drivers in making knowledgeable decisions about whether it is safe to overtake another vehicle on the road.The model can give drivers real-time feedback and alerts when it detects risky overtaking situations by utilizing several algorithms and live data from dash cam. So, from the intensive learning tendency to the implementation of the system, the noteworthy objectives of our research are:

1. To understand the details of the proposed algorithm by which the system will be implemented.

2. To deeply understand and to have a clear knowledge about vehicle detection and classification techniques i.e. CNN, Mask-R-CNN, convLSTM, YOLO versions etc

3. To understand and find an efficient way to image processing and using deep learning methods for determining parameters like speed, distance of vehicles

4. To implement convincing outcomes, to ensure that the system have constant performance in regular as well as complex situation

5. To improve traffic safety and lower the likelihood of accidents brought on by dangerous overtaking techniques.

6. To prevent simultaneous overtaking tendency of vehicle drivers.

7. To make drivers accustomed for maintaining speed limits.

8. To make our system user friendly and to use it in the autonomous vehicle industry with further improvement.

1.3 Future Scope

Autonomous system considers communication between vehicles to decide safe overtaking within minimum time. As the approach is designed for complex traffic situations where oncoming and ongoing both vehicles are running on two lane roads, they can decide whether they should overtake or not through communication by this autonomous system. The system will not have any high cost sensors or cameras, so in future, users will get this autonomous system in their vehicles at low cost. Moreover, speed maintenance where required, will be ensured as the system will assist users with high accuracy. This proposed method will make drivers accustomed to safe overtaking and maintaining speed while driving vehicles. Also the chances of road accidents will be decreased as the driver will be notified frequently about the distance between other vehicles. Besides, the driver will be getting feedback about overtaking in daylight and night-time also. So, the main focus of this paper is to determine safe overtaking of vehicles considering certain factors and by implementing them with image processing to prevent road accidents. The certain factors include speed and distance of associated vehicles, vehicle classification, lanes, detour point etc.

1.4 Chapter Outline

The structure of the thesis paper is as follows. We give an overview of the current condition of road safety and the difficulties that drivers encounter when passing other vehicles in chapter 1. In chapter 2, we present a review of the literature on safe overtaking techniques and the contribution of AI to enhancing traffic safety. We go over the process utilized to create the safe overtaking mechanism in chapter 3. We provide the findings of our trials and assess the effectiveness of the safe overtaking system in chapter 4. We go over the results' ramifications and possible advancements for the safe overtaking system in chapter 5. The thesis paper is concluded in chapter 6, which is the last chapter.

Chapter 2

Related Work

A vehicle that relies on the combination of sensors, artificial intelligence and powerful processors to execute software is called an autonomous vehicle. From fully manual to fully autonomous, there are 6 levels of automation according to the U.S. National Highway Traffic Safety Administration (NHTSA). Our goal is to start the revolution of automated vehicles in Bangladesh by introducing a level 1 (ADAS) Advanced Driving Assistance System. ADAS may aid the driver with steering, braking or accelerating including rear-view cameras and other features. The role of ADAS is to prevent deaths and injuries by reducing the number of car accidents and other means by including features like lane departure warning, pedestrian detection, blind spot detection etc. Through recent researches of ADAS, various ways of road safety and safe driving research has been discovered which use various features and approaches like IoT based approach, control methods, image processing, deep learning etc.

2.1 Previous ADAS works

The documentation of paper [6] provides a comprehensive survey of the technical trends and developments in advanced driver assistance systems (ADAS) and autonomous driving with a focus on the algorithms used in self-driving prototype automobiles. The authors review the current state of the technology, including the various sensors, algorithms, and control systems used in ADAS and autonomous driving such as it can automatically stop the vehicle if any vehicle comes in front of an ego-vehicle. This uses Automated Emergency Braking System (AEBS) although the driver is in the driving seat. Sparse feature tracking methods help to track using multiple image frames, using camera and LIDAR sensor to detect objects are main features of the system. Using the optical flow algorithm of 6D motion, it can reduce traffic accidents also. The paper also discusses the challenges associated with developing ADAS and autonomous driving systems, such as sensor fusion, real-time processing, and the need for robust and reliable algorithms. The authors identify several key research areas, such as improving perception algorithms, enhancing the reliability and safety of autonomous driving systems, and developing new control systems that can handle complex driving scenarios. The paper also examines the regulatory and ethical considerations associated with ADAS and autonomous driving, such as liability, privacy, and cybersecurity. The authors argue that the development of effective regulations and standards is essential for the safe and widespread adoption of these technologies. Overall, the paper provides a valuable overview of the technical trends and challenges associated with ADAS and autonomous driving, highlighting the need for continued research and development to improve the safety and reliability of these systems.

The paper [4] provides a comprehensive overview of various pedestrian detection techniques for advanced driver assistance systems (ADAS). The authors highlight the importance of pedestrian detection in ADAS, as it is crucial for ensuring the safety of pedestrians and reducing the risk of accidents. The paper examines various approaches to pedestrian detection, including feature-based, machine learning-based, and deep learning-based methods. The authors discuss the advantages and limitations of each approach, along with their applications in real-world scenarios. They also compare and contrast different algorithms and frameworks used for pedestrian detection, such as the Viola-Jones method and the Deep Neural Network-based method. The paper provides a detailed discussion of the datasets used for pedestrian detection, including popular datasets such as Caltech, KITTI, and Cityscapes. The authors evaluate the performance of different algorithms using metrics such as precision, recall, and F1 score. They also discuss the challenges associated with pedestrian detection, including occlusion, pose variation, varying scales and lighting conditions. The paper presents the current state-of-the-art techniques for pedestrian detection in ADAS and highlights the need for further research to improve their accuracy by using 3D sensors and the development of more robust and efficient algorithms. Overall, the paper provides a valuable resource for researchers and practitioners working on pedestrian detection in ADAS.

Now, in the research paper [22], authors present a model where road accidents can be avoided by detecting objects from the vehicle track. They use Raspberry Pi Camera Module for detecting objects and Advance classifier for classifying the object. The approach involves using various IoT devices such as cameras, radars, and lidars to capture data and images from the surrounding environment, which is then processed using machine learning algorithms to detect and classify different objects on the road. The technology has the potential to enhance driver situational awareness and provide real-time warnings to avoid collisions. However, several challenges must be addressed when developing an object detection system using IoT and machine learning, including optimizing the algorithms for real-time performance, handling noisy sensor data, and ensuring privacy and security. Despite these challenges, object detection using IoT and machine learning has the potential to significantly revolutionize road safety by providing drivers with enhanced situational awareness and real-time warnings. This technology can also be integrated with other ADAS systems, such as lane departure warning and automatic emergency braking, to provide a more comprehensive safety solution.

On the other hand, in [20] authors describe a vision-based driver assistance system that uses computer vision techniques to detect and warn drivers of potential forward collisions and unsafe overtaking maneuvers. The system is designed to process images captured by a single camera mounted on the front of the vehicle and is based on a combination of image processing and machine learning techniques. The paper presents the system's algorithm and outlines the steps involved in detecting and tracking the position and movement of vehicles on the road. The authors explain how the system uses different features, such as lane markings, vehicle size, and motion cue obtained from optical flow, to classify vehicles into different categories. The system also uses an overtaking detection algorithm that analyses the behaviour of vehicles on the road to detect unsafe overtaking maneuvers. The authors evaluate the system's performance through simulations and real-world experiments and report promising results. Overall, the paper demonstrates the potential of vision-based driver assistance systems to enhance road safety by providing real-time alerts and warnings to drivers. However, for safe overtaking, knowledge about a vehicle's distance, position, velocity and also the gap between two vehicles can give the driver a 360-degree view which partially is covered by the work [18] which presents a vision-based automatic warning system designed to prevent dangerous and illegal vehicle overtaking also. The system is based on a combination of computer vision techniques and machine learning algorithms and is designed to process real-time video streams captured by a camera mounted on the front of a vehicle. The authors describe the system's methodology, which includes several steps, such as image pre-processing, vehicle detection, and overtaking detection. The driver is given safe overtaking advice based on measuring the distance, calculating the vehicle velocity, and the frame rate of images of the front car from the ego-vehicle. The image pre-processing step involves reducing noise and enhancing image contrast to improve the accuracy of subsequent processing steps. The vehicle detection step uses a combination of deep learning algorithms to identify and track the position of vehicles on the road. Then, the vehicle will use data from the front camera to calculate distance using the triangle similarity method. The authors were also able to detect road markers for vehicles traveling in the opposite direction. The vehicle detection step uses a combination of deep learning algorithms to identify and track the position of vehicles on the road. The overtaking detection step involves analysing the movement patterns of vehicles to detect unsafe overtaking maneuvers. This system is built with Python, OpenCV, and Flask. The authors evaluate the performance of the system through simulations and real-world experiments, comparing the results to those of other state-of-the-art methods. The paper demonstrates that the proposed system is effective in detecting dangerous and illegal overtaking maneuvers and can provide real-time warnings to drivers. Although, the paper provides a valuable methodology for developing vision-based automatic warning systems to prevent dangerous and illegal vehicle overtaking, demonstrating the potential of computer vision and machine learning techniques to enhance road safety, it does not identify the vehicles coming from the back and does not calculate the cars' velocity. It will cause complications if the back vehicle likewise seeks to overtake.

Authors of [27] approach for both image processing and deep learning models to advise the driver for safe overtaking decisions. For both of these models, video from Remote Vehicle 1 (RV1) is used as input. The first phase in image processing models is to detect obstructions in the lane, followed by feature extraction and a judgment made to the driver depending on some manual setup conditions. However, the vehicles must register with a service in order for RV1 to transmit the decision to the Host Vehicle (HV). Whereas in the CNN model, videos of cars traveling are gathered, and the difference of frame (DOF) is analysed to determine whether or not a vehicle is safe to overtake. Despite the fact that less bandwidth and computational power are required in the image processing model, the model may fail if the server cannot handle real-time requests. Moreover, the CNN model could give us false results in unfamiliar situations.

In [15] a driver warning system for overtaking vehicles is introduced for both day and night time considering an overtaking condition. Using RCNN in daylight the overtaking vehicle is detected but in night due to lacking of light the overtaking vehicle is being detected using headlights if position and movement of vehicle meets predefined conditions. Authors use manually annotated recorded images as feeding data for their experiment. Even though their experiment shows satisfactory result the system simply detect the vehicles and the overtaking decision is totally up to the driver himself.
The other articles [3], [26] the vehicle driver gets assisted for overtaking in the night time. [3] aims to introduce a new robust vehicle detection and tracking method irrespective of the light and road conditions regardless of distance, using vision and sonar sensors. Authors propose two methods for detecting images: feature based and appearance based. The proposed system first detects the vehicle by feature-based approach and then cross checks with appearance-based approach. Using the upper environment and shadow region of a real time image, authors set the threshold value which indicates the day or night. Using light classification, they classify lights of roads, headlights, taillights, brake lights and reflected lights at night-time. Besides, for tracking they have used Lucas-Kanade Algorithm (LKA) using online templates and the Degree of Trust (DOT). On the other hand, [26] uses the help of taillight and headlight V2V distance, gap distance, measured speed and fuzzy conditions are applied according to that.
Paper [38], proposes a novel optimal trajectory generating scheme for autonomous vehicle overtaking that is both smooth and safe and can be used in a variety of traffic scenarios. The scheme is based on an optimal predictive problem and can be implemented in real-time. The proposed plan is based on the optimal predictive problem solution and aims to reduce driving expenses while lowering collision chances when there is any opposing vehicle in the overtaking lane. The technique has a nearly non-existent computational cost and supports real-time execution. Simulation results show that the scheme effectively obtains the optimal trajectories even in difficult overtaking contexts, and optimal overtaking costs are obtained for various states of the associated vehicles. The proposed technology can be employed as a fully automated system or an advanced driver assistance system (ADAS) to improve vehicle flows at challenging driving conditions and enhance transportation sustainability. On the contrary, the suggested technique has some drawbacks, such as the fact that perfect knowledge of the nearby traffic is required and that traffic flow uncertainties are not taken into account during the optimization process. Although the system can produce the best trajectories in emergency situations, any risky overtaking with the opposing vehicle in a hazard-filled environment should be done with an additional safety buffer. In order to combat the uncertainty and randomness of traffic, networked, automated cars can use V2V communication to exchange precise information rather than relying just on on-board sensors.

In the article [14] the suggested method solves the problem by establishing an ad-hoc connection with the vehicle to be overtaken in a 5G environment. The system is made up of a communication unit with a controller and a communication unit for determining when the vehicle can safely overtake. The antenna for vehicle-to-vehicle communication at 5G frequencies up to 20GHz in the proposed system is designed using fractal geometry. This allows the driver to safely pass larger vehicles without risking a traffic collision. However, depending upon the vehicle, complexity of the systems and various interfacing devices, installing vehicle to vehicle communication is costly. Considering V2V Communications, [12] has created an overtaking system model. Distance between front vehicle, driver's driving intention and vehicle status information help this system to analyze overtaking safety assistance. This system will also be able to give commands using fuzzy theory. Besides, for bad weather or for networking issues, a camera sensor may not be a choosing option whereas V2V wireless communication can play a vital role in this condition which is also ensured by this paper.

2.2 Lane Detection

Lane detection assists the car in maintaining its lane position. This technique can alert the driver if a vehicle is about to cross the lane for passing without signaling. Further enhancing the safety of overtaking operations, lane detection can also assist in informing the driver of impending road conditions, such as turns or crossroads. For achieving good results in lane detection, many researchers have proposed different algorithms and approaches over time.

For lane detection in driving scenes, the authors of this paper[17] mention a hybrid neural network that combines CNN and RNN. No matter the weather or the state of the roads, their primary goal was to detect the lane while driving. A network design based on an encoder-decoder structure was suggested. This system uses semantic segmentation to predict the lane of the current frame from a large number of continuous frames as input. The CNN encoder processes the input frames, creating a time-series of feature maps that represent the features that have been abstracted from each frame. The LSTM network receives these feature maps after which it learns the temporal relationships between them and forecasts the lane information for the upcoming frame. The CNN decoder then reconstructs the data and creates a probability map for lane prediction using the output of the LSTM network. In order to provide reliable lane detection in driving scenes, this architecture utilizes the advantages of both CNN and RNN in processing spatial and temporal information, respectively. Since they have used two different datasets their architecture delivers cutting-edge performance and is capable of managing complex situations. However, the computational efficiency of the architecture is not up the mark.

Another paper's [28] idea is to imply Probabilistic Hough Transform(PHT) algorithm with some advance parameters after enhancing and detecting the region of interest(ROI) from the input image.Basically,to prevent input images from being misled by other colors and to allow for greater detail, authors turned their RBG road images into HVS color images before applying grayscale and PHT.They highlight that the intelligent vehicle management system can benefit from their framework's integration into driverless cars. The usage of convolutional neural networks(CNN) is mentioned by both [25] , [23] and they both emphasize the value of preparation and data augmentation to enhance the dataset used to train the network. In [25], the

VPGNet deep learning model is used with the basic ZFNet network for improving the detection efficiency of the main model. The ZFNet's structure was also changed by the authors to better fit the objective of lane detection.

On the other hand, [23] gives a solution of excessive calculation and workload while detecting the lane with traditional approaches. They initiate an approach combining the cloud computing and edge computing since autonomous vehicles often need to process huge data in real time and they can reduce the time latency in this case. Additionally, they efficiently detect the lane line issue in multi-lane scenes and dynamically identify the current lane scene when vehicles change lanes by utilizing a convolutional neural network and semantic segmentation. However, one potential drawback from this could be network connectivity and cloud server availability. By directly getting the coordinates of lane line points from the drive scene, the proposed [16] end-to-end lane recognition method seeks to get beyond these drawbacks. With this method, great precision and real-time performance are achieved while removing the need for post-processing operations. The method is intended to be utilized with the integrated NVIDIA PX2 platform, which is frequently employed in automated driving systems. We are aware that each lane detection method has advantages and disadvantages of its own. In our situation, we focus on traffic line detection for recognizing the lane instead of road region segmentation. Also, deep learning always provides higher accuracy. Moreover, a compact network design with improved precision was required. Thus, we chose Point Instance Network (PINet) [19] for our research to determine the lane markings with lower False Positive. It has the capacity to identify specific lines on the road and precisely calculate their placements within the image regardless of the arbitrary traffic lanes. The basic idea of this method is to use a deep learning model inspired by a stacked hourglass network to forecast key points on lane lines and since it is relatively lightweight, making software integration a breeze. We will talk about its whole methodology in another section.

2.3 Semantic Segmentation

[8] In this paper the author used fully convolutional network (FCN), one kind of neural network for semantic segmentation which basically consists of convolutional layers. The author tried to improve this model by modifying skip connections, upsampling layers and so on. Moreover, the author used VGG16 network instead of fully connected layers with convolutional network. Though this paper helps us to get a starter knowledge about semantic segmentation but for large image and real-time applications, this method will be computationally expensive. To improve their efficiency, this paper also gives us some suggestions like pruning, quantization, knowledge distillation and so on. [9] Furthermore, if we look into this paper, then we will notice that the author used a deep learning approach named pilotNet which basically consists of convolutional layers. This paper is mainly based on calculating steering angles and speed of a vehicle from an image or real time scenario. Though this model can work well with bad weather also but there are also some improvement sectors like improving robustness, multi-model input, real time performance, transfer learning and so on.

[10] Besides, there is also a similar paper for semantic segmentation using fully convolutional network architecture with an importance-aware loss function. Basically, this paper introduced a model of semantic segmentation for autonomous vehicles which is computationally efficient. Though this paper broke the benchmark record of other semantic segmentation model but the development of more efficient real time inference can be a drawback of this paper.Moreover, the dataset did not cover real world driving scenarios which can also be another noticeable direction for future work also. [5]Adding depth channel to the input of the fully connected network and modifying pooling layers from depth maps, the author of this paper made another perspective of semantic segmentation model. Though this paper combines RGB and depth information together to improve semantic segmentation model but robustness to noise and occlusions, real time performance and integration with other modalities can be their improvement sectors.

[11] In this paper, the author did not focus on making a semantic segmentation model but the author used this model in a 3D object to improve the accuracy and it improved the accuracy noticeably. Moreover, this paper [13] gives us information about a model called, DenseASPP which is a semantic segmentation model and its full form is dense feature extractor with an Atrous Spatial Pyramid Pooling. Basically this model helps us to extract multi-scale context information. Though it gives us a new window in semantic segmentation but testing on difficult images, increasing computational efficiency, merging with other models and increasing accuracy in segmentation result was the future work of this paper. [30] If we want to look at a novel model for semantic segmentation then [30] this paper can help us. The author proposed a new model called FasterSeg for real time semantic segmentation. Taking input from a real time image, this model labels each point on different classes like ground, vehicle,building etc. This model also used a fully connected network (FCN) consisting of several convolutional layers. Working on missing, noisy and incomplete images, introducing with new environment were the future work of this paper. If we want to get an overview for all proposed models for semantic segmentation till now, then this paper [37] can catch our attention strongly. In this paper, the author talked about different models like U-Net, SegNet, DeepLabv3+, and PSPNet. Among all of them,DeepLabv3+ gave the best accuracy. Though DeepLab3+ can be an efficient model but this model needs large amount of test dataset. PSPNet also can be an efficient model but it needs high computational resources. In this way, through this paper, the author tried to provide us informations about different models of semantic segmentation.

Besides, in this paper**nafis9**, considering the challenges like low light, night driving, robustness to different weather, the author proposed a new model using irregular convolution and solved those challenges through this model for autonomous driving. Though this model can solve these problems but evaluation is limited to a few benchmark datasets and also this model was not tested in real world vehicles also. [39]In this paper, the author proposed a new model for semantic segmentation which is based on explainable AI and Adaptive Dehazing Approach. Here, explainable AI is used for image classification and Adaptive Dehazing Approach is used for removing haze from the image.This paper introduces a new novel model which can also outperform in bad weather also.

2.4 Monocular Depth Estimation

Moreover, there are several papers on monocular depth estimation which help us to get a clear and strong knowledge to apply it in our research. First of all, [31] this paper proposed a new novel approach for monocular depth estimation. Taking advantage of the concept of isometric self-sampling, which entails projecting the estimated depth values onto the image plane and contrasting them with the original image and finally, compute a reconstruction loss using these informations. Considering the improvement of robustness of depth estimation in challenging surfaces and applying this model to other computer vision tasks was the future work for this paper as mentioned.For extreme bad weather, [36] this paper introduced a new model using a monocular camera for depth estimation. Combining stereo matching algorithms and deep learning, this model generates a dense disparity map which basically helps to estimate the depth. Compared with other models, this model gives better accuracy and robustness but adding more sensors will give a more perfectful result which is mentioned as future work of this paper.

Considering the urban areas, [34] author proposed a new monocular depth estimation model. The novelty of this paper is using a refined U-V disparity mapping algorithm which helps to generate a disparity map. Though this model gets the higher accuracy in urban roads but considering the large urban environment, using this model to other autonomous systems was the future attention of this paper.Furthermore, [35] through this paper, the author tried to introduce a new methodology that enables UAVs to navigate autonomously in dynamic situations using monocular depth, optical flow, and ego-motion estimates. Besides, Geometric guiding is also included in the system in order to increase navigational accuracy and robustness. Additional geometric models for network refinement at test time can be a future attention for this paper. [33] The author suggested a novel method through this paper that uses a deep learning model which combines a geometric restriction between the size and distance of the vehicle to estimate its distance. In addition, the authors suggested using a new dataset called VD-CAM for testing and improving the suggested model. But the suggested framework needs to be thoroughly assessed under various road conditions for future work.

Moreover, [32] this paper gives us an overview of the challenges facing in calculating monocular depth estimation for unmanned aerial vehicles. In addition, the author also tried to discuss the merits and demerits of each method and also their limitations. Finally, the author offers a thorough analysis of the state-of-the-art methods for monocular depth estimate using deep learning for UAVs and suggests a number of exciting lines of inquiry for further study in this area.
Taking input as a 2D image and giving output of implicit depth estimation as a 3D image representation is the main goal of this paper [37]. The author of this paper, approached a new model for monocular 3D object detection that learns implicit depth estimation and showed that this model did really well compared to others models. The future indication of this paper is that it needs synthetic data for training which is the limitation for real-world scenarios.

Chapter 3

Proposed Method and Technology

Methodology— The implementation of the proposed method, used for two laned highways, in this paper is given in figure 3.1. In the simulation, vehicle HV (Host Vehicle) wants to overtake the vehicle TV (Target vehicle). In this case, there can be stationary and/or dynamic obstacles on both lanes like pedestrians or stalled vehicles. Our entire system will work with a single camera and an output screen. System will take input which is live videos from the dashboard camera of the HV. The taken input is then processed inside a processing unit installed inside the HV. We used four deep learning models where the processor will process all the necessary methods to generate the overtaking assisting information to help HV to make the accurate decision. After getting the trained model we detected and classified the vehicles firstly.Lane detection was also done in parallel here. Then, using detection algorithm, we applied semantic segmentation for depth estimation and optical flow algorithm for velocity approximation.Then after taking the information of distance and velocity, we give a suggestion whether the driver can do overtake or not.

The suggestion of overtaking will happen or not will be depend on three things mainly. If our target vehicle's speed is lower than the Host vehicle's speed.It will basically mean than If we accelerate the Host vehicle more it can cross the Target vehicle. After confirming this, we will check if our OV is in the safe distance. Finally if we have enough space before our TV, our model will suggest to Overtake.

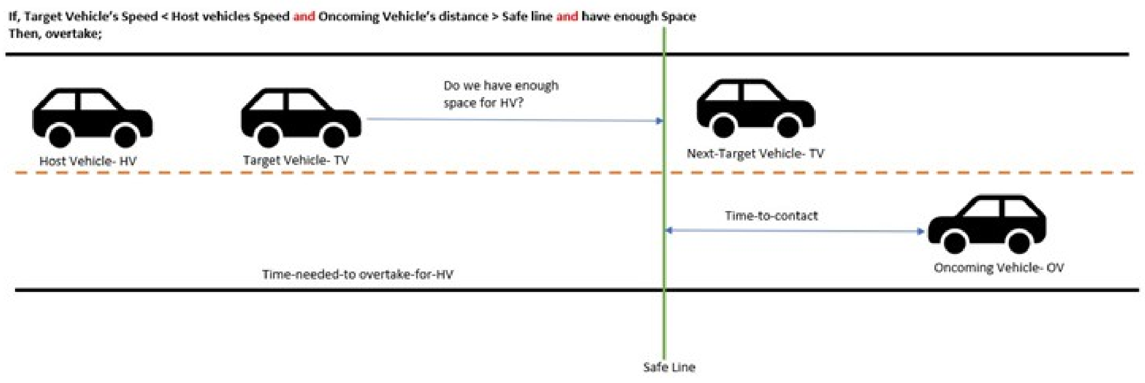
If, Target Vehicle's Speed < Host vehicles Speed and Oncoming Vehicle's distance > Safe line and have enough Space
Then, overtake;
Do we have enough space for HV?
Host Vehicle- HV
Target Vehicle- TV
Next-Target Vehicle- TV
Time-to-contact
Oncoming Vehicle- OV
Time-needed-to overtake-for-HV
Safe Line

Figure 3.1: Overtaking Process

In the proposed model, we used deep learning based approach which is comprised of the following steps:
1. Data Pre-processing (Collection, noise cancellation, splitting, training)
2. Vehicle detection and classification using YOLOv5
3. Lane Detection with PiNet
4. Parameter Estimation
5. Decision Making

3.1 Data Pre-processing

3.1.1 Data Collection

Finding a proper data set for the proposed model to feed was one of the prominent problems we faced. As our thesis is on overtaking system decision in Bangladesh, our first job was to collect the raw images of Bangladeshi's vehicles. Many sources offer vehicle dataset, but they are not appropriate for our country. As in Bangladesh various type of vehicles are found for the detection, we need to define all of them. At last, the dataset we used the Mendeley Data website's dataset called, "Poribohon-BD" version2 which offers the help for various purposes [21]. This vehicle dataset consists of 15 native vehicles of Bangladesh. The vehicles are: Bicycle, Boat, Bus, Car, CNG, Easy-bike, Horse-cart, Launch, Leguna, Motorbike, Rickshaw, Tractor, Truck, Van, Wheelbarrow. However, We took 5 type's vehicles in our custom data set which are bike, bus, car, truck and bicycle as, we are working on highway road.

Along with these, we have used 5 Videos of Day Time, 5 Videos of Night Time and 5 Videos of Rainy Night. We have tested these videos in our merged final model. We collected these videos from several places. Most of them are from our own vehicle's dashcam while driving in the highway of Bangladesh, mostly from Dhaka-Comilla Highway. Some videos are taken from YouTube videos which are recorded in Bangladesh highway. Lastly, we had to take some foreign videos because we were unable to find enough videos of Rainy Night. Here is the tabled statistics:

Type of Video Data	Number of Video Data
Daylight Video	5
Night Video	5
Rainy-Night Video	5

Table 3.1: Primary Testing Data

3.1.2 Dataset Annotation

There are 9058 photos in the collected data overall, with a wide range of stances, angles, lighting, weather, and backgrounds. The pictures are all in JPG format. 9058 image annotation files total are included in the dataset. A Multi-class vehicle dataset is also available. These files list the precise locations of the items in the accompanying image that have labels. The annotation was done manually in the collected data, and the values that were annotated are kept in XML files. The photos have been labelled using the LabelIng tool by Tzuta Lin. Additionally, data augmentation methods have been used to maintain an equivalent number of photos for each type of vehicle. Also blurred to preserve privacy and secrecy are human faces.However,To form our usable dataset, we took help from “Roboflow”. “Roboflow” is such kind of online platform in where we can build our custom dataset with our images and their corresponding annotations. There are annotations for each image which contain the xml file for that corresponding image according to our model.

3.2 Vehicle detection and classification using YOLOv5

For object recognition, Ultralytics has created a cutting-edge deep learning model called Yolov5 (You Only Look Once version 5). In this section, we will discuss the YOLOv5 technique in further depth, including the model’s architecture, Optimization function, Loss function and its vehicle classification method used to determine the network’s efficacy.

YOLOv5 is a step up in both precision and velocity over its YOLOv5 predecessor. In order to efficiently learn feature representations, YOLOv5 employs a ”backbone” known as deep neural network architecture which is based on the EfficientNet model. The model also makes use of a number of additional characteristics, such as anchor boxes and feature pyramid networks, to enhance its precision. The increased speed of YOLOv5 compared to YOLOv4 is one of its most notable upgrades. There is also a lot of algorithms which can detect vehicle i.e. Mask R-CNN, Faster R-CNN, SSD etc but we find YOLOv5 more suitable from those algorithms for our system. If we compare YOLOv5 with Mask R-CNN, Mask R-CNN may struggle to detect small objects that are close together but in this case, YOLOv5 can perform satisfactorily. Not only that, Mask R-CNN is only capable of processing 11 frames per second whereas YOLOv5 can process up to 155 frames per second on a GPU. In comparison, YOLOv5 is also better than Faster CNN in terms of critical speed and efficiency, such as in real-time object detection applications. Through this comparison, we decided to use YOLOv5 to detect and classify vehicles. Also it is a one-stage detector that detects and classifies objects in one network pass. In general,

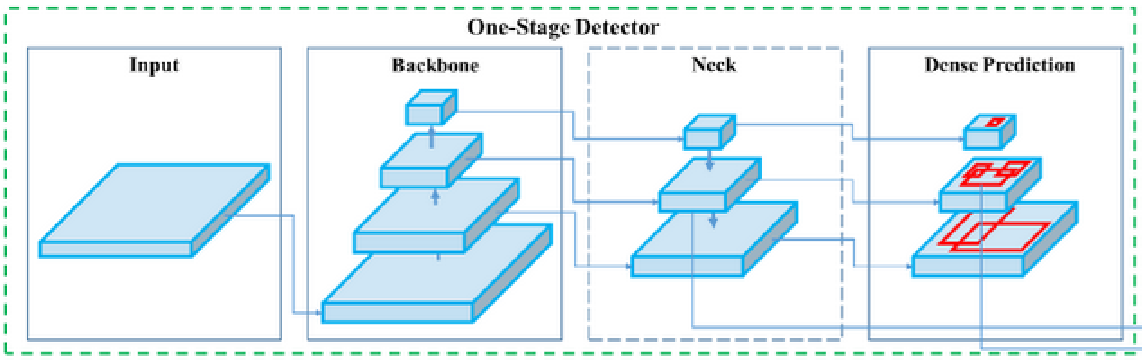

Figure 3.2: One-Stage Detector

YOLOv5 is a superior option for identifying moving autos owing to the fact that it is faster, more accurate, and less complicated. The model's excellent accuracy and recall rates for vehicle detection make it useful to us in a wide range of settings, including traffic monitoring, parking management, and autonomous cars. Not only vehicle detection but also categorization of vehicles are performed simultaneously in real time using YOLOv5. So if we have a picture with several items, we find YOLOv5, which can find them all with only one run across the network. In object identification benchmarks like COCO and VOC, the Yolov5 model has performed at or near the top. Surveillance, robotics and autonomous vehicles are just some of the real-world uses for this technology.

3.2.1 YOLOv5 Architecture

There are three primary components of the Yolov5 method, and they are the "backbone network," "neck network," and "head network," respectively.

Backbone network: Deep convolutional neural networks (CNNs) like ResNet, EfficientNet, and CSPNet are frequently used as the backbone network. After processing the input image, a collection of feature maps are generated that accurately represent the image's most salient characteristics. CSPDarknet53 is a modification of the Darknet53 architecture used in YOLOv3 and forms the basis of YOLOv5's backbone network. In order to maximise network performance and accuracy, CSPDarknet53 employs a number of techniques, including channel splitting, shortcut connections, and convolutional layers. The architecture is shown in figure 3.3:

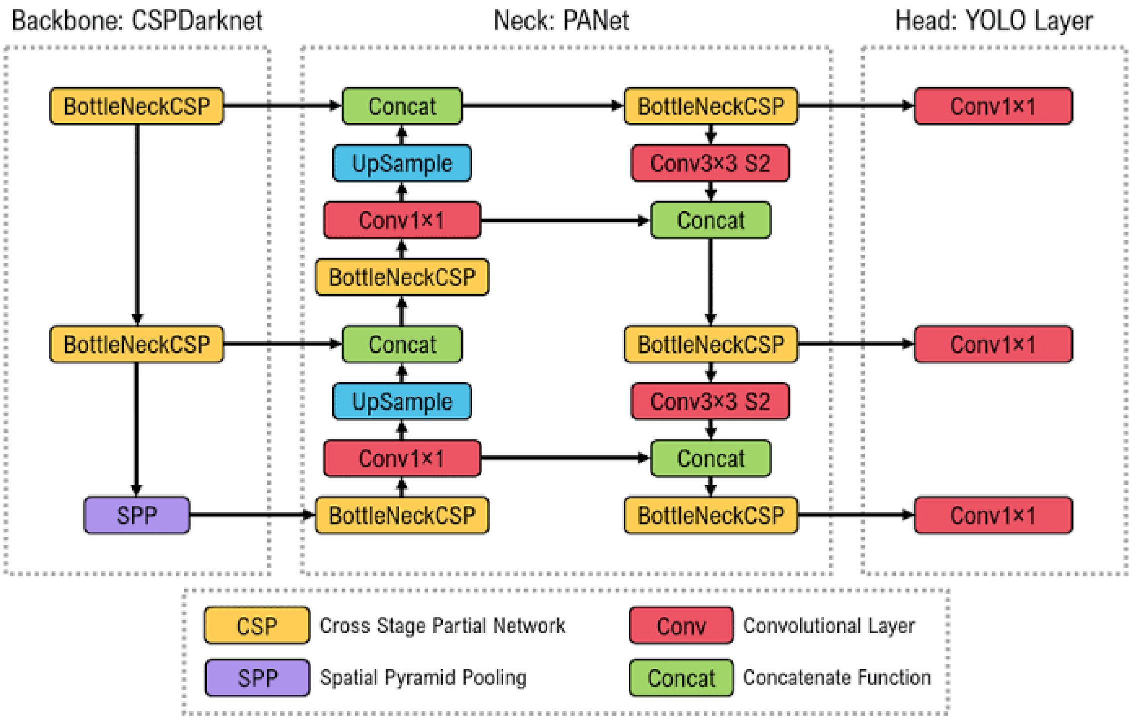

Figure 3.3: YOLOv5 Architecture

Neck network: The neck network consists of many intermediary layers that carry out supplemental feature extraction and fusion. Using its feature pyramid network

(FPN) as a neck network, the algorithm of Yolov5 can recognize objects of varying sizes and shapes. The fundamental concept of FPN is to generate a multi-scale feature pyramid by constructing a top-down pathway and lateral connections from a backbone network. This is done in order to accomplish this goal. The backbone network is often made up of a deep convolutional neural network, also known as a CNN. This type of network works to analyse the input picture and derive a collection of feature maps at varying sizes. In YOLOv5, the feature maps can be combined in three different ways to create three distinct neck networks such as SPP (Spatial Pyramid Pooling) neck, PAN (path aggregation network), CSP (Cross Stage Partial).

Head network: The bounding boxes and objectness scores for each grid cell are predicted by the head network. Yolov5 makes predictions about the position and size of the bounding boxes by utilising anchor boxes, and it employs a sigmoid function to make predictions about the objectness score of each box. In addition, a classification layer is added to the YOLOv5's head network to aid with vehicle categorization. In order to further categorise the observed items, such as vehicles, trucks, buses, etc., the classification layer is utilised.

3.2.2 Optimization function of YOLOv5

The well-known stochastic gradient descent (SGD) technique provides the foundation for YOLOv5's optimization function, which combines momentum with the algorithm. To be more specific, YOLOv5 makes use of an optimizer called Adam, which is an extension of SGD that computes adaptive learning rates for every parameter in the neural network.

Whenever the Adam optimizer is used, the neural network's weights are revised by the following formula:

$$w(t+1) = w(t) - lr * m(t+1)/\sqrt{(v(t+1)} + \epsilon) \tag{3.1}$$

where w(t) represents the weights at time t, lr stands for the learning rate, m(t+1) represents the first moment estimate (also known as momentum) at time t+1, v(t+1) represents the second moment estimate at time t+1, and eps is a tiny number that prevents division by zero from occurring.
During the training phase of the neural network, the Adam optimizer is responsible for computing the gradients of the loss function with respect to each parameter in the network, and then updating those parameters in accordance with the results of those computations.

3.2.3 Loss Function

To prevent YOLOv5 from becoming overly confident in its bounding box and objectness score predictions during training, the loss function employs a penalty for getting these metrics wrong. Localization error, classification error, and confidence loss are all included in the loss function to encourage the model to produce reliable predictions. At the time of inference, YOLOv5 will use non-maximum suppression, also known as NMS, to the predicted bounding boxes in order to eliminate overlapping detections and obtain the final collection of identified objects. Through this, the algorithm is able to distinguish between a broad variety of items, such as automobiles, people, animals, and others.

3.2.4 Vehicle Classification method

There has to be a collection of annotated photos of cars to train YOLOv5 for vehicle classification. Images of several automobiles, taken from a variety of perspectives and in a wide range of lighting situations, should be included in this set of data. It is possible to train the YOLOv5 model with localization loss and classification loss once the dataset is ready. YOLOv5's head network, in particular, provides an output of bounding boxes and class probabilities for each identified item in the input picture. Each object's possibility of belonging to a certain class is represented by its class probability.

3.3 Lane Detection with PiNet

Several autonomous driving systems depend on lane detection to enable safe navigation and to offer critical information about the structure of the road. A neural network design called PINet (Point Instance Network) creates points on lanes and separates expected points into individual instances. The PINet methodology will be covered in detail in this section, along with the network's architecture, the training loss functions, and the evaluation metrics that are used to gauge the network's effectiveness.

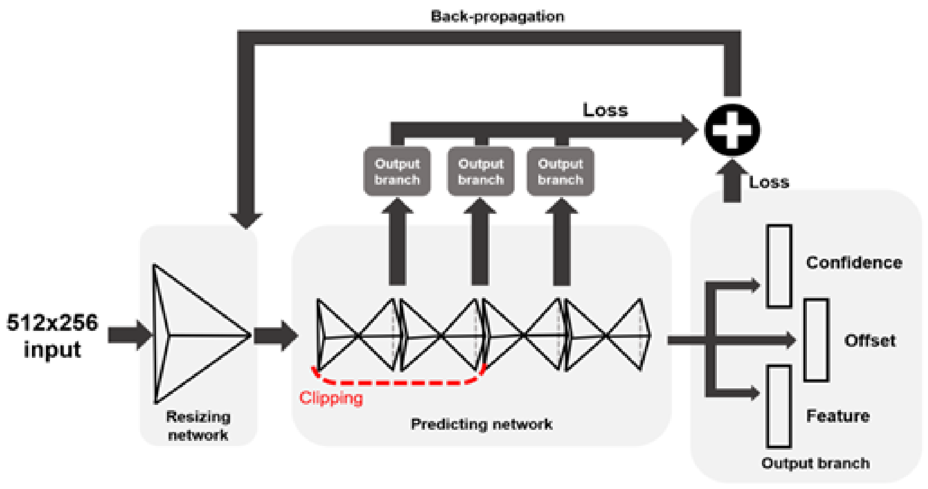

Figure 3.4: PiNet Architecture

3.3.1 PINet Architecture

From figure 3.4, we can see that the three fundamental parts of PINet are the resizing network, the forecasting network, and the output branch. To begin with, we gave the resizing network our RGB input video frames so that it might shrink the video frames as images. For instance, a 512 x 256 RGB image can be compressed into a 64 x 32 image depending on the convolutional layer sequence. The prediction network, which consists of stacked hourglass modules which we can see in figure 3.1, appears after that. Four hourglass modules are employed in our research. After scaling and preprocessing the input image, the predicting network carries out a series of operations to extract features and generate predictions. Mainly, the hourglass modules are responsible for predicting the points on the lane marker. The output branch, which forecasts the confidence, offset, and embedding characteristics, is then applied for clustering the points on the lane and smoothing it. Embedding output branch is responsible for instance segmentation for every lane. However, the author from [21] finds that, when light levels are low, PINet predicts the lane incorrectly, like it predicts the lane on the off-road. As a result, they made the decision to build ROI using the road mask produced by SegNet segmentation outcome. Thus, it helped to isolate the roads to which the PINet algorithm would be applicable.

In more detail, to conserve memory and compute time, the resizing network reduces the size of the RGB input image. It has three convolution layers, each of which has a filter of 3x3, a stride of 2, and a padding of 1. Activation of the PReLU and batch normalization are performed after each convolution layer. The resizing network produces a resized image with a 64x32 resolution as its output.

The predicting network uses the scaled image generated by the resizing network to forecast the precise locations of the traffic points and embedding properties, such as segmentation. A number of hourglass modules, each with an encoder, decoder, and three output branches, make up the predicting network. The encoder and decoder are made up of several bottleneck modules with skip connections that send data from different scales to deeper layers. Three different sorts of bottleneck modules are available: same, down, and up. The output produced by the same bottleneck equals the size of the input. A convolution layer with a filter size of 3, stride 2, and padding 1 is used to replace the first layer of the down bottleneck in the encoder, which is used for down sampling. The up bottleneck, which employs a transposed convolution layer with a filter size of 3, stride 2, and padding 1, is used for upsampling in the decoder.

Each output branch of the prediction network, containing three convolution layers, yields a 64x32 grid. The output branches project the levels of confidence for the presence, offset, and embedding of the important points in each cell of the output grid. With a different channel for each output branch, the corresponding loss function is applied in line with the goals of each output branch. Hourglass modules can be added to the prediction network in any number, and they are all simultaneously trained using the same loss function. After the training period, we can pick on our own how many hourglass modules to use in accordance with the processing power available.

Fig 3.5 is mainly showing that a video frame is giving to the Resize Network which compressed to smaller version for better prepossessing and memory efficiency. Then via the hourglass Network PINet is predicting three values where Confidence and

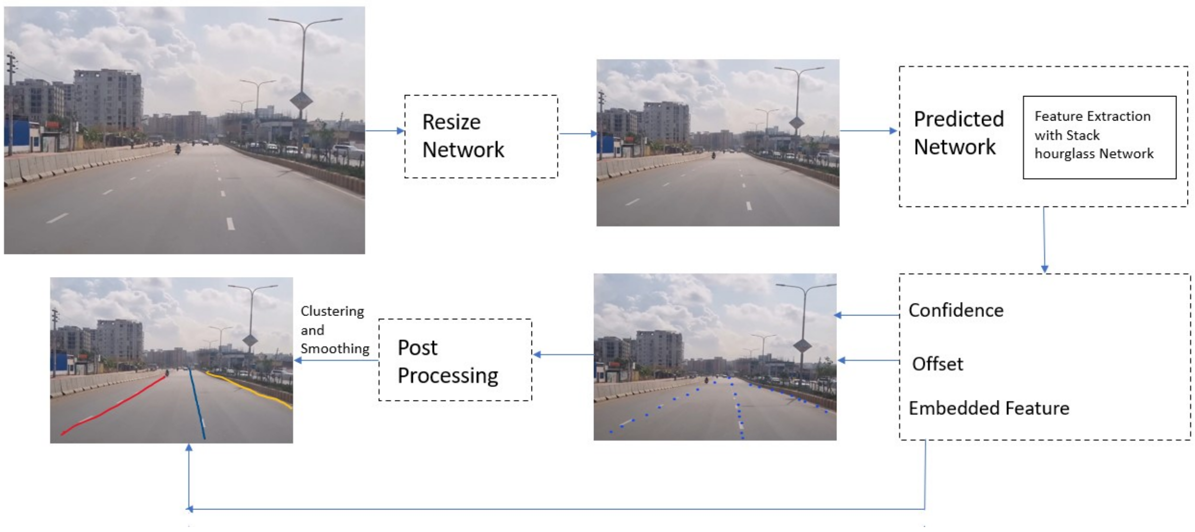

Figure 3.5: The workings of PINet

Offset value is predicting the points on the lane. Finally with the help of embedded features and more processing we got the output of predicted points into each instances.

3.3.2 Loss Function of PINet

Four loss functions are applied to each output branch of the hourglass networks in order to train the PINet network. The specifics of each loss function are as follows: The confidence output branch, which forecasts the presence of a lane line, is subjected to this loss function. The prominent object identification algorithm YOLO (You Only Look Once) served as an inspiration for the confidence loss function. The binary cross-entropy between the anticipated confidence value and the ground truth label is used to determine the confidence loss.

Offset loss function is used to estimate the offset of the lane line from the center of the grid cell. It is applied to the offset output branch. The L1 distance between the projected offset and the actual offset is added to determine the offset loss.
The embedding output branch, which creates the embedding characteristics for each predicted point, is subjected to this loss function. The clustering procedure uses the embedding characteristic to identify each instance. The SPGN (Similarity Group Proposal Network) instance segmentation method, which presents a straightforward procedure and a loss function, served as the inspiration for the embedding loss function.
So the overall loss from this function is:
$L_{total} = \gamma_e L_{exist} + \gamma_n L_{non-exist} + \gamma_o L_{off\ set} + \gamma_f L_{feature} + \gamma_d L_{distillation}$

3.4 Parameter Estimation

To acquire the decision regarding overtaking, some parameters like velocity, acceleration, distance needs to be calculated. In our system we calculated both distance for the detected target vehicle and relative velocity from the host vehicle. For the host vehicle, speed from the speedometer will be analysed. Speed of the vehicle coming from the other lane will also be calculated in this case. We used two different methods to pursue our calculation and to get more accuracy. The distance is determined from a single camera using the semantic segmentation labelling. Then, velocity is estimated collaborating the optical flow with deep learning YOLOv5 model.

3.4.1 Segmentation Using SegNet Model:

In this computer vision age, semantic segmentation is really a novel initiative to label each pixel of an image so that it can identify the object and region for a particular area. Basically, semantic segmentation means dividing the image into several segments. For clear vision, if we think about a street image containing vehicles, people, road and sky then dividing each of the areas by combining each pixel for a particular object (like vehicles, people, road etc.) is called semantic segmentation. A mask image is obtained as output from a semantic segmentation model which labels each object into different colors.This method is very useful for observing any scenario, dividing objects in a particular frame, autonomous driving and so on. This whole method can be divided into two parts. One is for segmentation and another is for calculating or estimating the depth. In this case, segmentation is done by supervised learning method and depth estimation is done by self-supervised learning method. For supervised segmentation, mainly, it is a training procedure which helps the image to classify each label. The training procedure shows the output of labeled images for different objects with the help of taking a dataset with annotation. After training the model, it can be used to label each pixel with appropriate semantic category. Furthermore, Self-supervised monocular depth estimation is an approach to calculate the distance relative to the camera of each pixel from a single image. It is a novel method to make a conversion from 3D image to 2D image from a single camera image and as it is self-supervised, so it can be trained on any image sequence. To find the distance between two vehicles using less hardware, this method is very effective.

As mentioned before, we have used four models in our system. Semantic segmentation and monocular depth estimation were two of them. Semantic segmentation is used for classifying or labeling roads and vehicles for our system. In addition, using those labels, we calculate the distance between two vehicles in front of us using a single camera.

There are two types of segmentation for image classification. One is semantic segmentation and another is instance segmentation. In semantic segmentation, the same type of objects are classified with a single color. In addition, the same type of objects are classified with different colors or labels. For example, if two or more cars are detected in an image, then each car is labeled with different colors in instance segmentation but for semantic segmentation, all the cars are labeled with a single color. In our system, we have used semantic segmentation.
First we need the bounding box for a particular object in an image (in our case, these are vehicle, road, sidewalk etc.). For this we need a proper edge contour detection that will segment and label all the objects per frame. In this model, the output goal is to provide the same size of image with a label of each object. The

approach we have used here is SegNet. It uses the encoder-decoder architectures to develop a neural network.

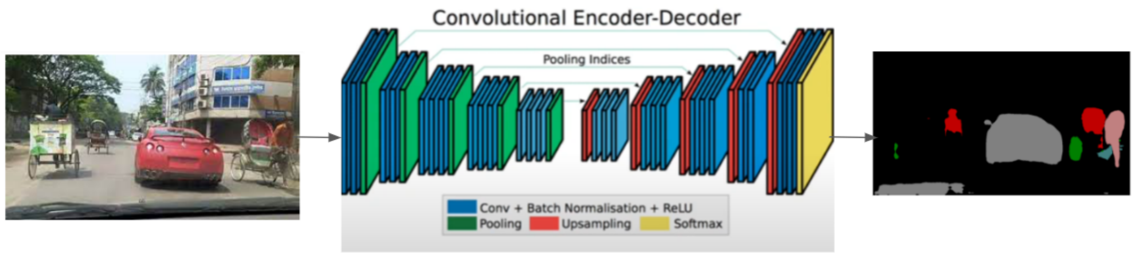
Convolutional Encoder-Decoder
Pooling Indices
Conv + Batch Normalisation + ReLU
Pooling
Upsampling
Softmax

Figure 3.6: SegNet Model Architechture

The first part of this method consists of both convolutional network layers and max pooling layers. This method also uses batch normalization layers and value activation functions also. Basically, the upsampling layer makes this model better than others. The convolutional and pooling layers are in the encoder part. In addition, convolutional and upsampling layers are in the decoder part. This model also used convolutional filters and their parameters to get better accuracy. Transposed convolutional part of SegNet is the new addition of this model. As there are both large and small size objects, so we need to increase the feature map. Besides, We suggest a grouped average pooling with three distinct kernel sizes: 2 by 2, 3 by 3 and 5 by 5. For every down sampling layer, it is paired with an up sampling layer. In this way, using this model, we segment the objects in every single image.

3.4.2 Distance Measuring Method:

In this segment, we will discuss which approach we have chosen for calculating distance between our vehicle to others. We have used one single camera for the input capturing and from that we segmented each frame and calculated the distance. As we have used a single camera, that's why it is called monocular depth estimation approach. We used a unique hierarchical method for measuring distance using a monocular camera that makes use of the outputs from semantic segmentation and YOLO. There are two main steps to our process. The scene's main point of interest is perceived during the first stage. In order to determine the precise distance measurement, the second stage involves a more thorough object examination. As the SegNet model keeps the image size same in output also, we need to remove the background and crop the object only to calculate the distance. As can be seen in Fig. The segmentation mask and depth map of the identified object are therefore cropped and also remove the background using the bounding box coordinates. The cropped mask also contains some background pixels. So, to remove those pixels, we used min pooling approach with the kernel size. Min Pooling =

$$I^{H\times W} \longrightarrow I'^{\left|\frac{H}{K}\right|\times\left|\frac{W}{K}\right|}$$

Here, H, W and K represent height, width and kernel size respectively.
Furthermore, we also replace the zero pixel to NAN values so that background depth points can be removed more perfectfully. Lastly, we also used Gaussian Noise Removal method so that noisy depth points could not affect distance calculation. The final step is to global average the resulting depth points to get the distance of all surrounding objects.

3.4.3 Speed Estimation Using Optical Flow

Optical Flow has many applications including structure from motion, Video Compression, Video Stabilization, object tracking and detection etc. However, We aimed to use optical flow as our velocity estimation approach. We can determine the pattern of image objects between two successive frames that is responsible for object movement with the help of optical flow. Each vector in a 2D vector field represents the displacement of a point from one frame to the next. To solve the problem of optical flow, we have to assume pixel intensities in two consecutive frames are the same. If a pixel I(x,y,t) in the first frame moves by distance (dx,dy) in the next frame over time dt time difference, there is no modification in non-changing intensity and the pixels remain the same. So,

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \tag{3.2}$$

Another assumption from optical flow is neighboring pixels having the same motion. It implements Taylor series approximation of the right-hand side, eliminating any common terms, and dividing by dt.

There are two types in optical flow where Sparse optical flow spots some certain pixels in a frame and Dense optical flow returns all the pixels known as flow vectors from the entire frame. Dense optical flow, as a matter of fact has the higher accuracy but of course costing higher computational cost. As for our work, we need to use certain interesting features from the input images or video frames, so we used Sparse optical flow regarding the computational cost also.

Sparse optical flow firstly chooses some important or interesting features such as edges in the object to track over time. Then, extracted features are run through the optical flow function to calculate the optical flow or the motion. To detect the best interest in a frame Shi-Tomasi corner detector is used which is also very similar to another popular method named Harris Corner Detector. Shi-Tomasi method is more effective than the other one because of a small modification they made. Both methods determine windows with large gradient and then compute the large corner response function score, R for which each window is classified as a flat, edge, or corner.Here comes the effective modification Shi-tomasi has made. They simply proposed, for the points R is greater than a threshold, will be the points counted as corner. In the Harris corner Detector, the scoring function is given by,

$$R = det\,(M) - k\,(trace)^2$$
$$det\,M = \lambda_1 \lambda_2$$
$$trace\,M = \lambda_1 + \lambda_2$$

Whereas Shi-Tomasi proposed the scoring function as,

$$R = min\,(\lambda_1, \lambda_2) \tag{3.3}$$

All of these are covered in [2].These method is implemented as goodFeaturesToTrack() in OpenCV. Again, Lucas and Kanade proposed an effective technique to estimate the motion of interesting features by comparing two consecutive frames in their paper [1] Which is also implemented by OpenCv as calcOpticalFlowPyrLK(). So we can easily use this functions from opencv to our implementation in next.

The main methodology is to merge YOLOv5 and optical flow.To get the velocity we followed three steps: Detect, Track and Calculate Distance vector where the detection will be done using our YOLOv5 model. Firstly we use YOLOv5 to detect objects in each frame of a video sequence. Next, by applying optical flow we can estimate the motion of the detected objects between frames. This motion information can then be used to refine the object detection results in subsequent frames. To track we pass the YOLOv5 output into Optical Flow and measure the distance vector for each pixel. Lastly the velocity is determined from the distance vector using the proper time unit. Overall, the combination of YOLOv5 and optical flow can provide a powerful tool for object detection and tracking in video sequences, and can be applied to a wide range of applications such as surveillance, autonomous driving, and robotics.

Chapter 4

Implementation

In this chapter we thoroughly describes the way we implemented our methodology and analyse the accomplished results. Our entire system will work with a single camera which will be placed on Dashboard and an output screen. Again, now a days most of the vehicle have inbuilt dashcam. We processed the system decision through a CPU which is compatible for the raspberry pi module. Then, our system works with a certain methodology and the implementation goes as following. As we have already mentioned, we used four different deep learning models. After data Pre-processing we trained the previously split dataset for YOLOv5 custom object detection model. At first we detect the vehicles which are in front of us through a detection model. Then we pass that detection box or coordinate to semantic segmentation model which gives us the exact coordinate of vehicles. After that, we pass those coordinates of vehicles to the depth estimation model to measure the distance of other vehicles which are in front of us. After that we calculate the relative velocity also of those vehicles using optical flow. In addition, we also measure the lane so that the driver can understand if he is on the right track or not.Lastly, we checked all the logical view points before giving the decision using another algorithm. In this way, measuring all of these model's output, our system will help the driver to take overtaking initiatives.

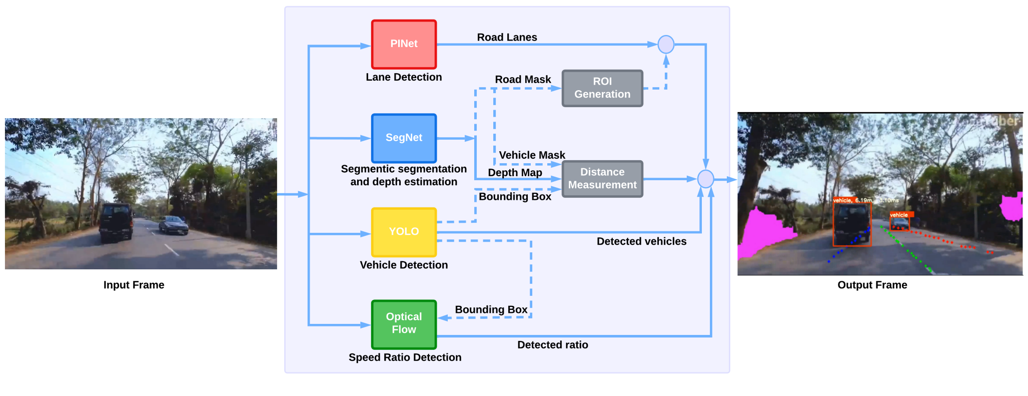

Figure 4.1: Implementation Process

4.1 Data Pre-prossessing

Using “Roboflow” we also divided our custom dataset into train, test and valid part. For making our custom dataset, we have used YOLOv5 model and with the model expected image size. So, our final usable dataset was customized after these steps.

Data Segments	Percentage	Total images
Training	70% of the total data	700
Validation	20% of the total data	200
Testing	10% of the total data	100

Table 4.1: Dataset splitting

4.2 Vehicle detection and classification

In this model, one of the primary tasks is to detect and classify the vehicles. To make our proposed model compatible with native vehicles, a custom trained YOLOv5 classifier is used here through the following steps:

4.2.1 Training the Model and Noise cancellation

For the training part, we have checked with different epochs values. The more we increase the epochs value, the more we get the increased accuracy in detecting vehicles. Lastly we have used epochs=100 as for not having better GPU. After training, we test the model’s performance on the video to detect and classify the vehicles by Using a video processing library to extract frames from the video and apply the YOLOv5 model on each frame to detect and classify the vehicles. We also applied pre-processing techniques and classification thresholding to remove duplicate detections, refine the bounding boxes and classify the detected objects. Then we integrate the model with other components and coordinate the detection and classification with next tasks. As we are working with only 5 objects which are bus, truck, motorbike, car and pedestrians, we set the name parameters to a list of their names. The list and the testing output is given below:

Name of Vehicle	Number of Vehicle
bike	112
bus	193
car	172
truck	153
Bicycle	70

Table 4.2: Numbers of Selected Vehicles

4.2.2 Model Testing

The figure 4.2 contains output images from video input and image input.First of all, we have made a video of a road in Bangladesh containing vehicles using the dashcam

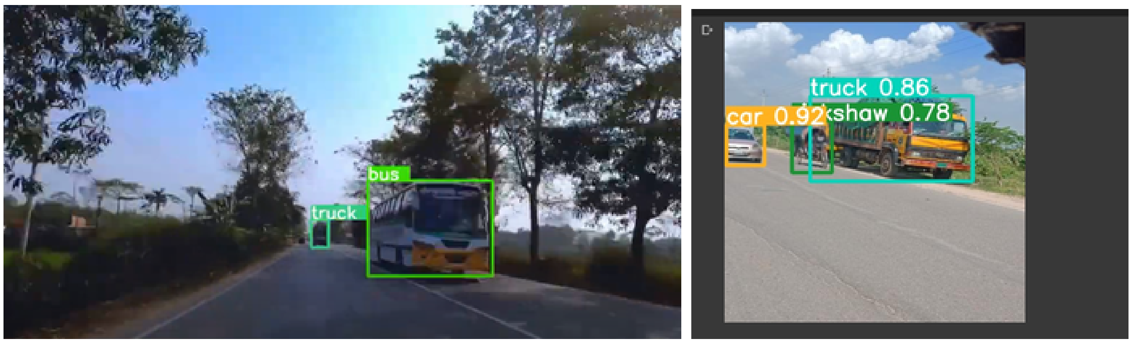

Figure 4.2: Vehicle Detection and Classification

of a car. Then we have extracted frames from the created video and labelled all the frames with bounding boxes around the vehicles. To train our model in terms of labelling vehicles, we have created a configuration file which contains video path, model architecture, training parameters, augmentation techniques, and classification labels. Here, we specified the number of classes that we want to detect and classify.

4.3 Lane Detection

As we mentioned before, numerous researchers have used lane detection to collect data on how autonomous vehicles perceive their surroundings. Using a deep learning based technique always offers a competitive result. Though typically it requires a lot of data to train the neural network models. As a result, we utilized PINet, which contains the hourglass stack and has a small infrastructure yet is a deep learning model. Our lane detection model is used to alert the driver if our front vehicle is leaving its lane or not. If the front vehicle is attempting to leave the lane, consider in our thesis that it might wish to overtake its front lane car.

For lane detection, we tested this model with three videos with a duration of around 30 Seconds. We made the videos from dashcam and mobiles with different pixel sizes. Since, PINet should be able to detect lanes from complex environments too , thus we selected videos from Day Light, Night, Rainy Night. Moreover, it can be trained regardless of the traffic lines.

This suggested technique is trained using the TuSimple and CULane datasets. All input video frames are downsized to 512 by 256 pixels and their RGB values are adjusted from 0 to 255 to 0 to 1. Some traffic lines, particularly those along the horizontal line, are only sparsely documented as a result of the annotation technique utilized in these datasets. To solve this problem, extra annotations are added to the x-axis every 10 pixels using linear regression on the original data. Several data amplification methods like shadowing, adding noise, flipping, translation, rotation, and intensity change are also applied. The output frames come with these modifications from the resizing network.

The predictions Network plays a vital role. It consists of several stack hourglasses. The number of our hourglasses was four which depends on the fact of the computing power of the target environment. These four stack hourglasses are trained with the

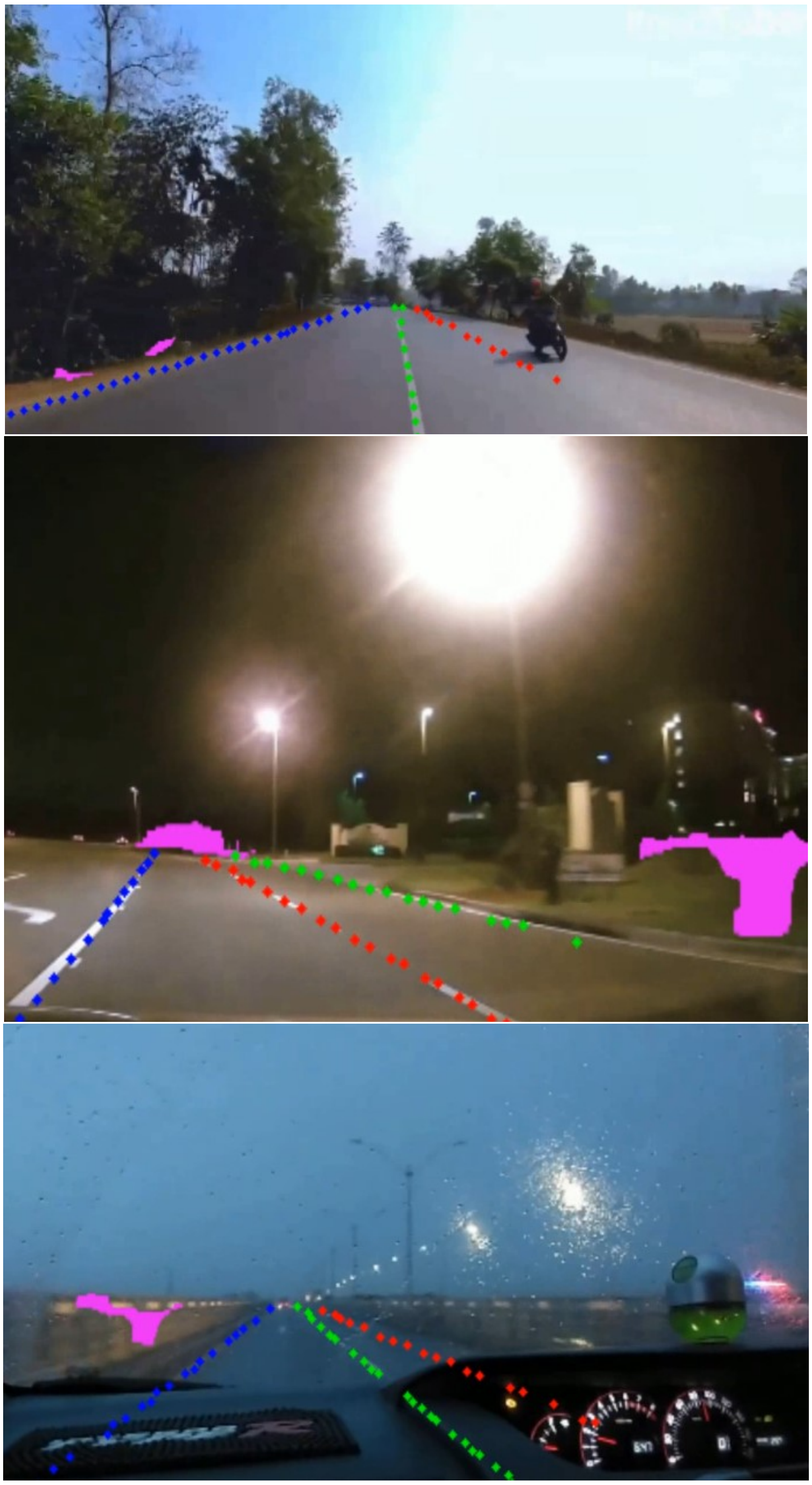
Figure 4.3: Lane Detection in Day, Night and Rainy Night

same loss function. The encoder and decoder part are used to downsample and upsample our video frames. After maintaining high quality feature maps,it starts the key point estimation in the video frames by the prediction networks. In the end, with the help of embedding functions we got the clusters of points on the lanes. The PINet forecasts the exact position of critical spots on traffic lines, and the spline curve fitting approach is utilized to generate a smoother curve. On the TuSimple and CULane datasets, the method's performance is assessed, with fair efficiency and low false positive rates observed.

We have used PyTorch code for a neural network that uses PReLU activations rather than ReLU activations in the bottleneck layers to implement the UNet design. For image segmentation tasks, the UNet architecture is frequently employed because of its efficiency.

Meanwhile, we faced some challenges like predicting the lane on the off-road. We searched for a solution to handle these issues, we found that ROI from road masks can help to identify the highway lane only. We applied our SegNet model for road masking here, the same model which we have used for vehicle masking. Thus we merged these two models together .However, we found some other wrong predicting lanes and outliers. So In PINet various hyperparameters need to be properly tweaked for optimal performance. For example, we set acceptable values for the learning rate, weight decay, and dropout rate. Additionally, we experimented with various regularization methods and activation function types.

All things considered, the PINet implementation entails meticulous dataset preparation, model architecture design, hyperparameter tuning, and model training on GPU hardware(GTX1060). The assessment procedure comprises the use of evaluation metrics to assess the accuracy and robustness of the model on diverse datasets. The performance of the model is enhanced by the proposed method's inclusion of additional data augmentation and imbalanced data handling techniques.

4.4 Distance and Velocity Estimation

4.4.1 Distance Approximation

As we want to calculate the distance of other vehicles which will be seen in the output screen, we divided the task into two subparts. Initially, using Semantic Segmentation, we take every classified frame of the video which is classified by the object detection model as the first input of this model. Then each frame is segmented into different objects using RGB.

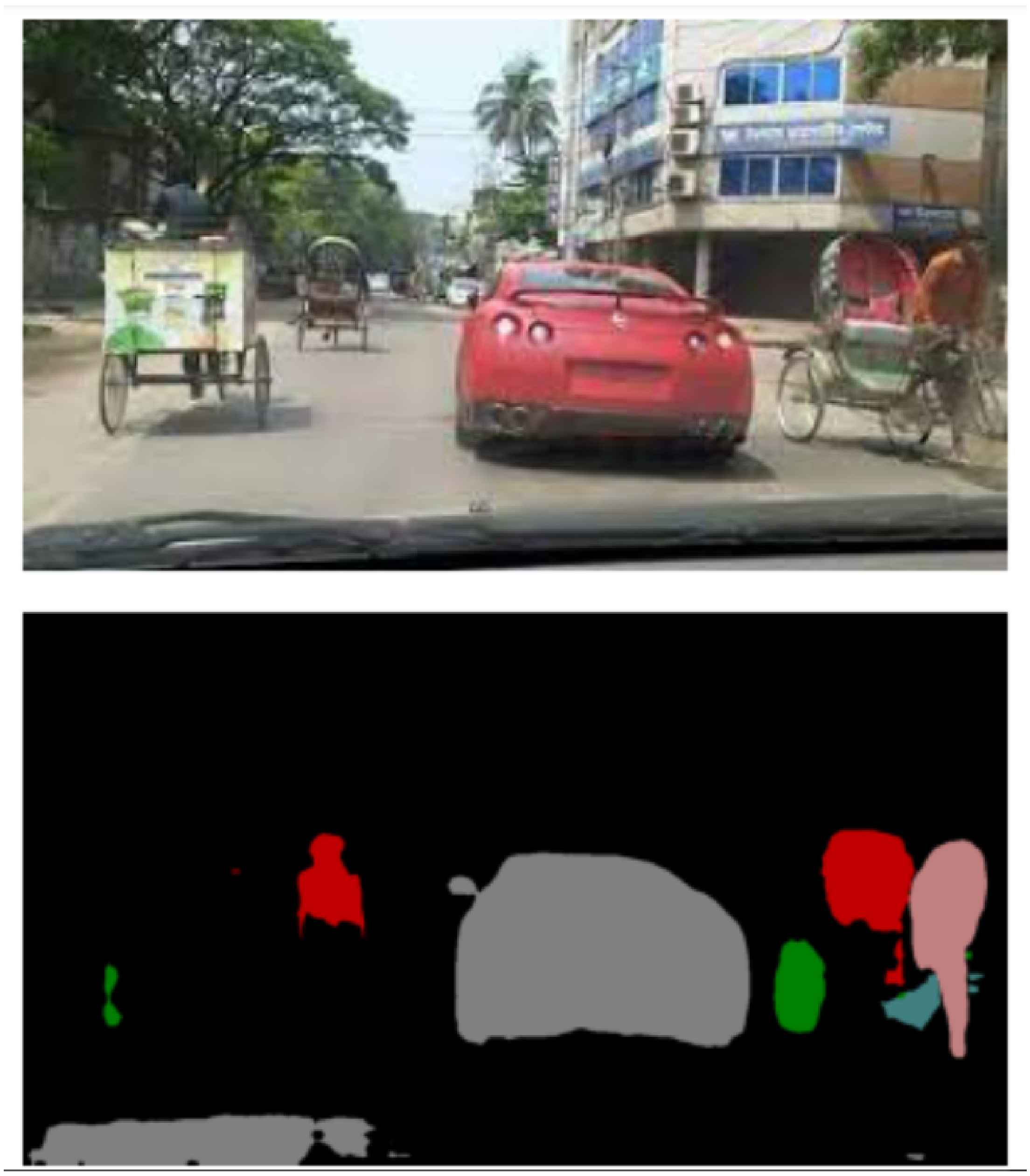

Figure 4.4: Image Segmentation With RGB

We have used the SegNet model for our system. Basically with the help of encoder and decoder, we create classification layers. These classify layers work by pixel wise calculation. With the help of 16 convolutional networks, the classification layers are divided. There is a functionality called max pooling which helps to upsample the feature maps or the classified objects. After doing some component extraction, we calculated the depth maps of every vehicle. We trained this model with BDD100k dataset to map the vehicles.

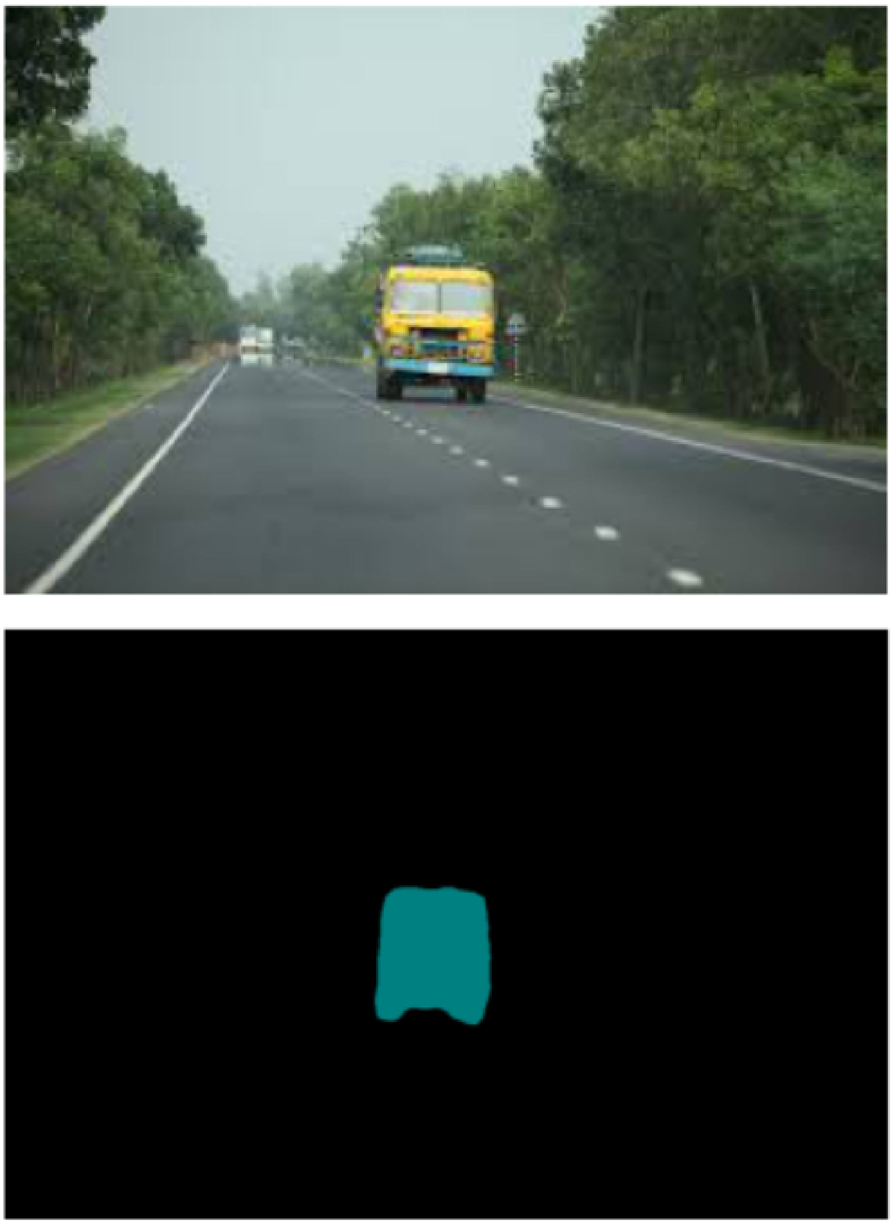
Figure 4.5: Depth Estimation

Basically this is the process of the depth estimation model. In addition, there were some methods for the gradient scale layers and also some methods for pose estimation. Both of them were used for better performance. By following these steps, we are finally able to crop the vehicle's depth map.
After using the SegNet model to classify each frame of our video, our next goal was to calculate the distance. To get a more accurate distance calculation, we had to process the classified vehicles to crop the vehicle's depth map and remove the background. Then we crop the vehicle portion of each image by doing some coordinate calculation as in figure 4.4:

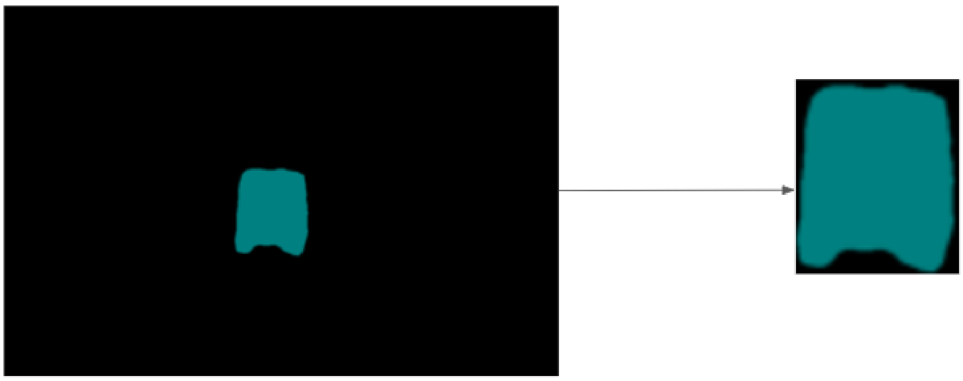

Figure 4.6: Vehicle Cropping Aproach

Some pixels of other objects were also included with our desired image. After using the min pooling method, we replace all those unwanted pixels to NAN values so that we can remove those pixels. After removing those pixels, there were also some noises in each frame. To remove those noises, we used a gaussian noise removal approach. After that we generated depth points from those depth maps. Lastly, averaging all of the depth points, we calculate the distance of all surrounding vehicles in each frame. It helped us to calculate the depth map more perfectly. If we want to make a step by step process for both of these two models, then it will look like below:

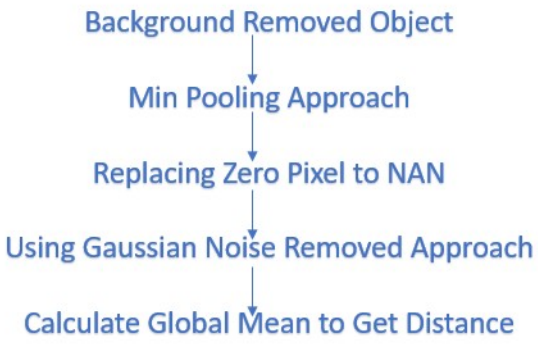
Background Removed Object
Min Pooling Approach
Replacing Zero Pixel to NAN
Using Gaussian Noise Removed Approach
Calculate Global Mean to Get Distance

Figure 4.7: Segmentation and Distance Calculation Process

4.4.2 Velocity Estimation

YOLOv5 is an object detection model that is used to identify and locate objects within an image or video frame. Optical flow, on the other hand, is a technique used in computer vision to track the motion of objects between frames in a video. In terms of velocity estimation, merging YOLOv5 and optical flow is one possible application which will improve the accuracy of object detection by incorporating information about the movement of objects over time. This helps to reduce false positives and false negatives in the detection process. To check our optical flow algorithm is workable or not, we run it on a test video to track the vehicles which is shown in figure 4.8. After detecting the vehicle using YOLOv5 custom-trained model. The model outputs bounding boxes that define the location of each vehicle in the frame.Then, we compute the optical flow vectors for each detected vehicle in the current frame relative to the previous frame. Then, use the optical flow vectors to estimate the position of each vehicle in the current frame. Speed is equal to the magnitude of the displacement vector between the previous position and current position of the vehicle, multiplied by the frame rate of the video. Lastly, we can change the pixels per frame output to real world unit by scaling factor from camera calibration or pixel ratio.

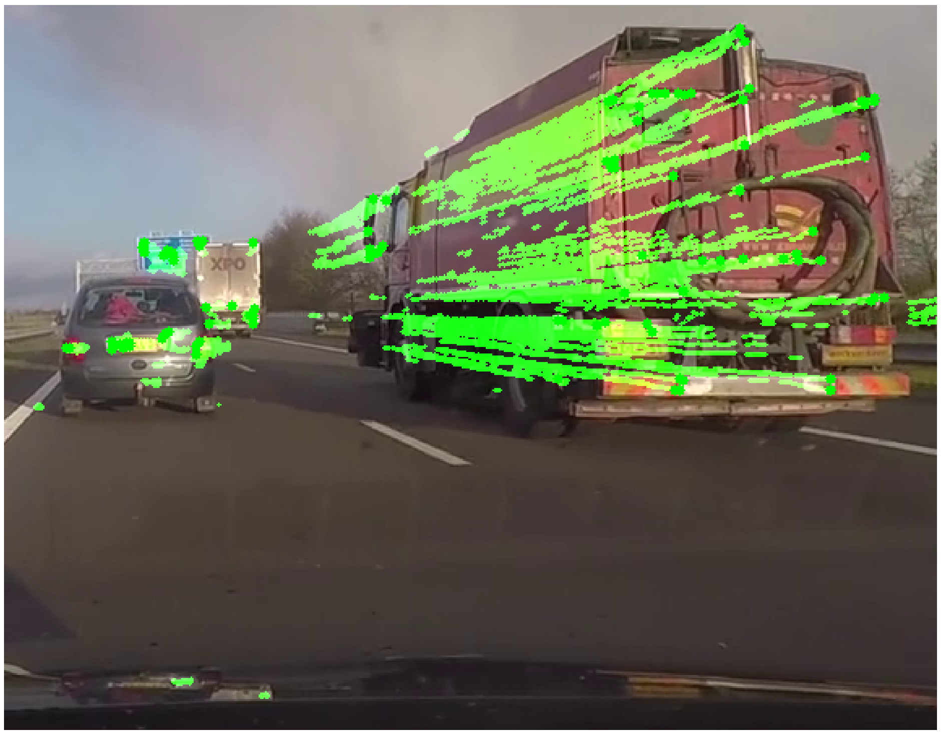

Figure 4.8: Tracking using Optical Flow

To calculate the displacement vector for each tracked vehicle conversion of the optical flow vectors from pixel coordinates to physical coordinates is done. The optical flow vectors obtained from OpenCV's function are in pixel coordinates. To convert them to physical coordinates, we need to know the scale factor, which represents the distance covered by one pixel in the real world. Here, this is estimated by Assuming distance-to-pixel ratio is 1:1. This can also be estimated using the camera calibration parameters. Now, the displacement vector is simply the difference between the current position and the previous position of the vehicle. The frame rate is obtained using OpenCV's property. Lastly, to convert the velocity to real world unit we used the following formula for the conversion:

$$speedToMetersPersecond = speedToPixelsPerFrame * \frac{distanceToPixelRatio}{frameRate} \tag{4.1}$$

4.5 Decision Algorithm

As we have mentioned earlier, our model starts working through a video input. After that it will detect the vehicles, segment them and finally calculate the distance and velocity. Therefore, the model can predict the overtake decision.

There are some significant conditions to predict the decision. We have set a threshold distance which is 9 meters. Besides, we have set another threshold distance which is 6 meters because we know that a large vehicle like a truck is almost 6 meters long in our country. If the front vehicle is far away than 9 meters, then our model does not suggest any decision because the front vehicle is not in closer distance. Moreover, if the distance of the front vehicle is between 9 to 6 meters then our model will check if the relative velocity is decreasing or not. If the relative velocity is decreasing, that means the host vehicle's speed is more than the front vehicle and then our model gives the suggestion for making an overtaking decision. In addition, if the relative velocity is not decreasing that means the front vehicle's speed is increasing, then our model shows the "not overtaking decision" on the decision screen. Besides, when the host vehicle peeps at to see whether there is any vehicle in front of the front vehicle, then if the distance is smaller than 6 meter and also velocity is larger than the host car, then our model does not give the over taking decision. Last but not the least, if the front vehicle is not detected, which means there is no vehicle in front of the host vehicle, then our model does not provide any suggestion.

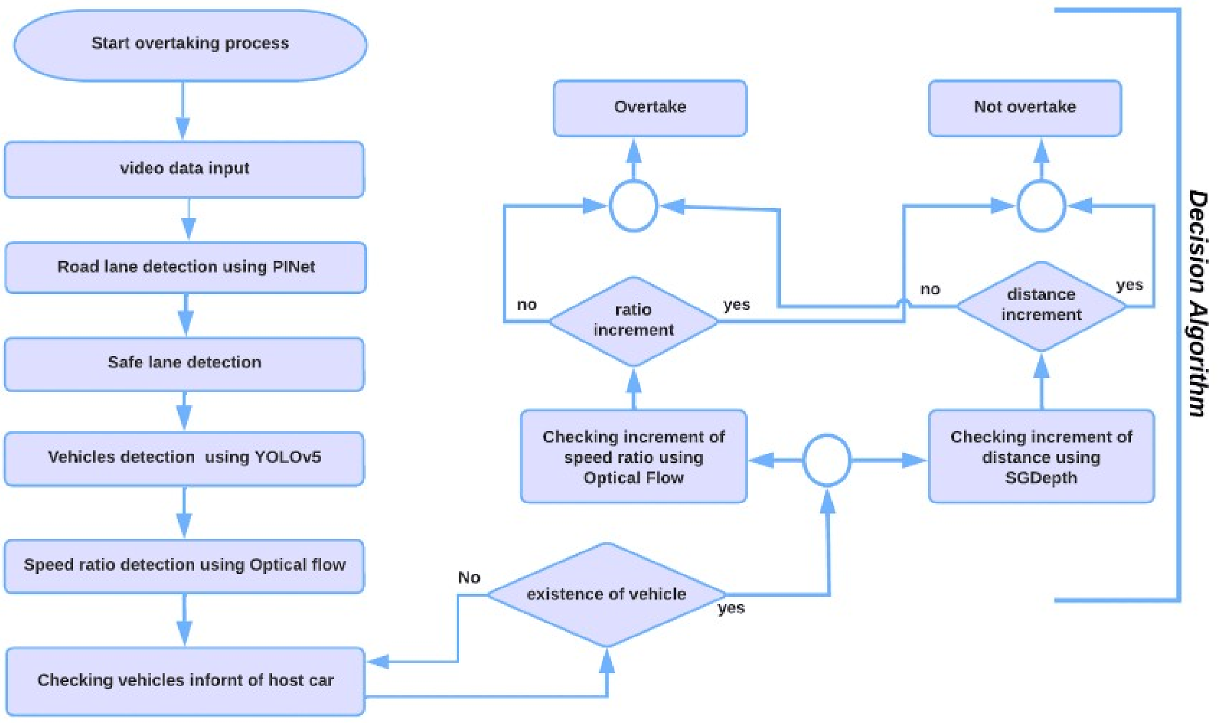
Start overtaking process
video data input
Road lane detection using PINet
Safe lane detection
Vehicles detection using YOLOv5
Speed ratio detection using Optical flow
Checking vehicles infornt of host car
No
existence of vehicle
yes
Checking increment of speed ratio using Optical Flow
Checking increment of distance using SGDepth
no
ratio increment
yes
no
distance increment
yes
Overtake
Not overtake
Decision Algorithm

Figure 4.9: Decision taking Flow

Chapter 5

Result Analysis

As we have merged too many models altogether to get the overtaking decision, so it is very important to check whether our main model gives the best decision or not.

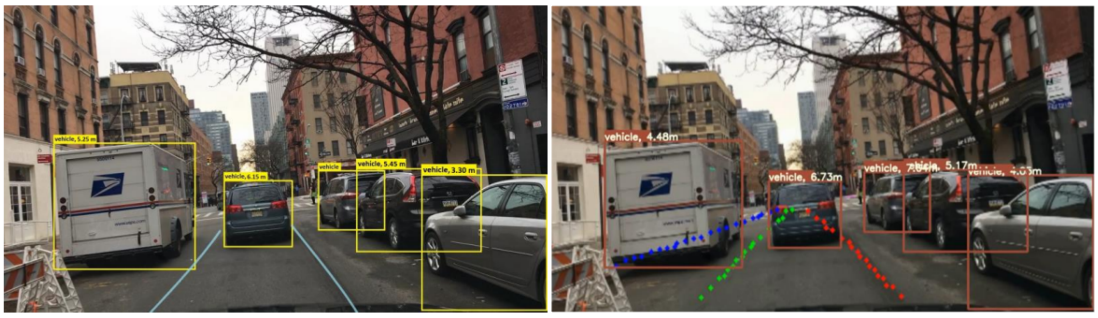

Figure 5.1: Ground Truth Of Distance Measurement Using BDD100k

We have used a semantic segmentation approach through the SegNet model to get the distance. We checked if the calculated distance value is similar to the actual value or not. For this part, we ran our model on both BDD100k dataset and our own datasets. Firstly, we ran our model on our datasets and measured the distance and then we measured the distance with the BDD100k dataset. As the distance is labeled in BDD100K dataset, we can notice that there is slightly difference in distance result. From different distances, we also measure the distance by using measuring tape on our own video datasets. Similarly, in this scenario, we have also noticed that the error is slightly more or less.

Here, we can see that there are 10 times, we count the distance using both our model and manually. In addition, we notice that the highest error is 0.62 meter and lowest is 0.37 meter. So, here we can also say that distance calculation is fair enough to get the overtaking decision. To measure the accuracy of our model from this table, we have used two types of metrics such as mean absolute error (MAE) and root mean square error (RMSE). Firstly, The MAE calculates the average size of the model predictions' mistakes. By averaging the absolute differences between the actual values and the anticipated values, it is calculated. The formula is:

MAE = $(1/n) * \Sigma|\text{Actual - Predicted}|$
Calculating the MAE value, we got MAE of 0.51 meters. Secondly, RMSE measures the square root of the average of the squared differences between the actual values

Table 5.1: Comparison of Actual Distance and Model's Distance Measurements

Count	Actual Distance (m)	Model's Distance (m)	Error (m)
1	4.27	4.89	0.62
2	5.50	5.91	0.41
3	7	7.53	0.53
4	8.54	8.91	0.37
5	9	9.61	0.61
6	9.5	10.12	0.62
7	10	9.56	0.44
8	12.82	13.43	0.61
9	17	17.39	0.39
10	21.5	21.99	0.49

and the predicted values:
Calculating the MAE value, we got MAE of 0.51 meters. Secondly, RMSE measures the square root of the average of the squared differences between the actual values and the predicted values:

$$R - squared = 1 - (\sum(Actual - Predicted)^2 / \sum(Actual - Mean)^2) \tag{5.1}$$

After calculating the R-squared value, we got R-squared = 0.76. According to these metrics, the model's error is estimated to be 0.51 meters on average, and the RMSE indicates that the model's predictions are typically accurate to 0.59 meters of the actual distance. The model explains around 76 percentage of the variance in the actual values, according to the R-squared value of 0.76, which is a respectable match for a model. Furthermore, we have applied those values into a graph diagram to visualize the result. Here we can see most of the cases, our model predicts slightly the wrong distance. Besides, only one out of ten times, our model predicts the distance which is less than the actual distance. In addition, it is noticeable that our model predicts the distance which is slightly more than the actual distance and so there will be a certain amount of extra distance which helps the driver to take any decision suddenly. It is also mentionable that the ups and downs of the model's result may occur due to camera calibration also. Moreover, as we have not trained the loss function, it can be another reason for not calculating the perfect distance.

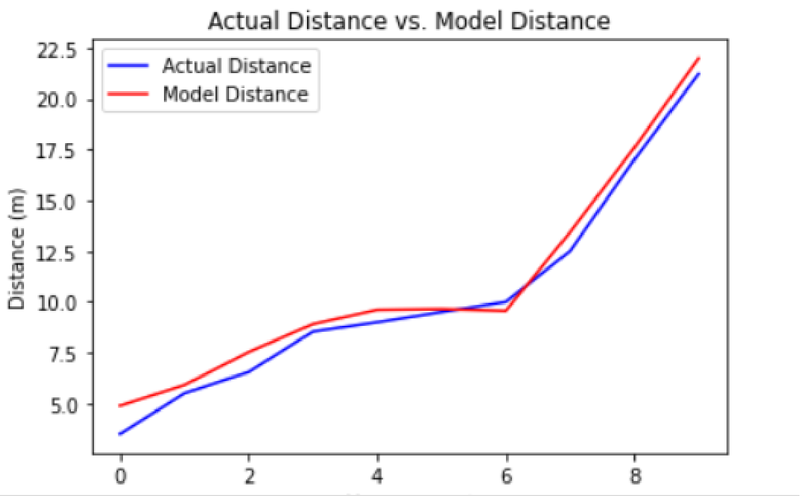

Figure 5.2: Ground Truth Of Distance Measurement

Now, for overtaking decisions, as our video data set is 1 minute 20 second long, so there are around 2325 frames. After getting the output we ran our video and checked manually whether it was a perfect overtaking decision or not. For this checking, we used google sheet to store the values. It took almost 1 hour to collect the information from 2325 frames. After completing the collection, we saw that 1551 times, our model detected that it was a perfect overtaking decision and the model detected a wrong overtaking decision for around 150 times. After that we noticed for the not overtaking moments, then we saw that there were 284 times when our model detected wrong for not overtaking decision but it detected perfectly 340 times for this perspective.

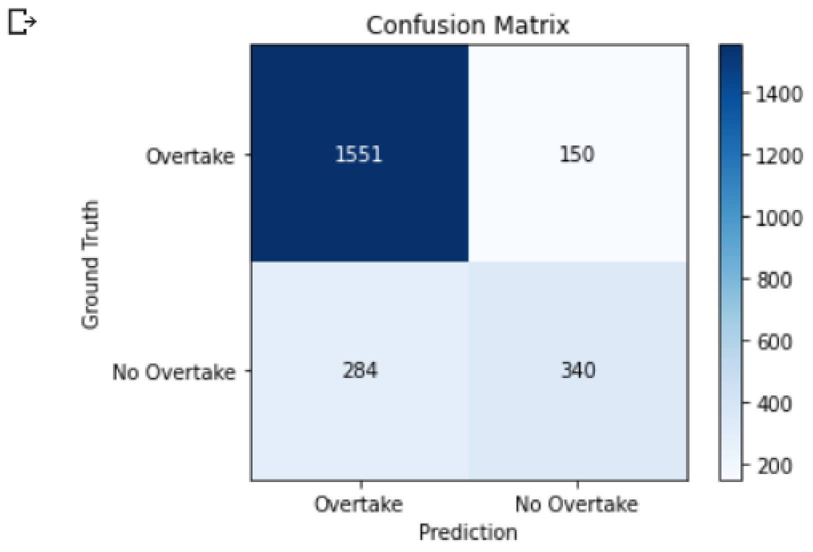

Figure 5.3: Confusion Matrix for Overtaking And Not-Overtaking Prediction

From the Fig.5.3 we can visualize how many times we got the true positive, true negative, false positive and false negative. But if we want to visualize the representation of accuracy, precision, recall and f1 score, then this bar chart will help us.

Hare, we can see that from 2325 frames, our model's accuracy is around 82 percentage. Moreover, we got true positive predictions out of all positive predictions, called precision which is around 91 percentage. In addition, our recall and f1 score is also good which are 85percentage and 88percentage respectively.

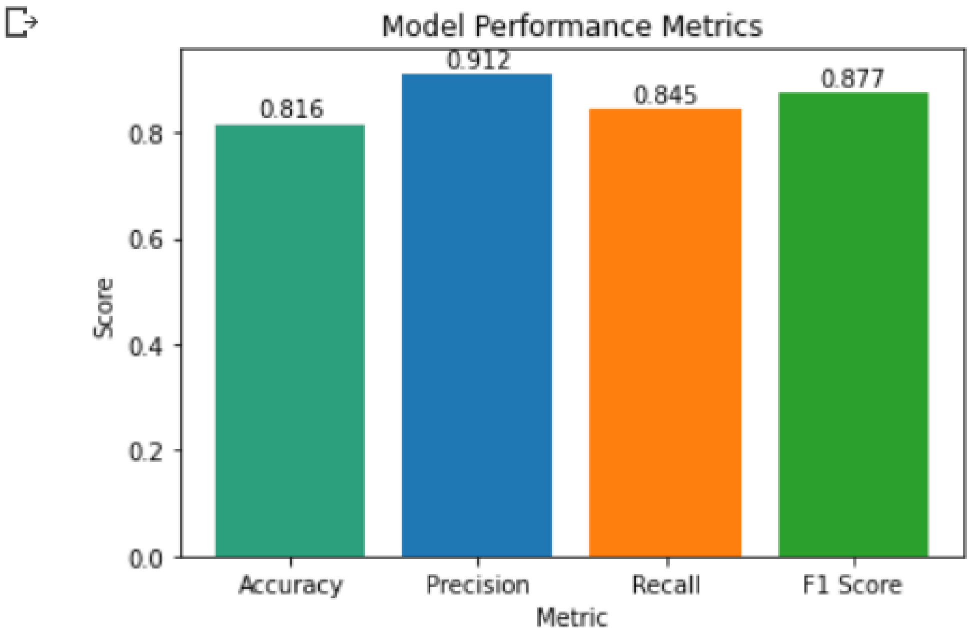

Figure 5.4: Bar Chart for Overtaking And Not-Overtaking Prediction

Here are some screenshots of our sample out video.
In figure 5.5, we can see that relative velocity and distance both are decreasing and host vehicle speed is increasing. That's why the drive can make an attempt to overtake and our model predicts the same thing.

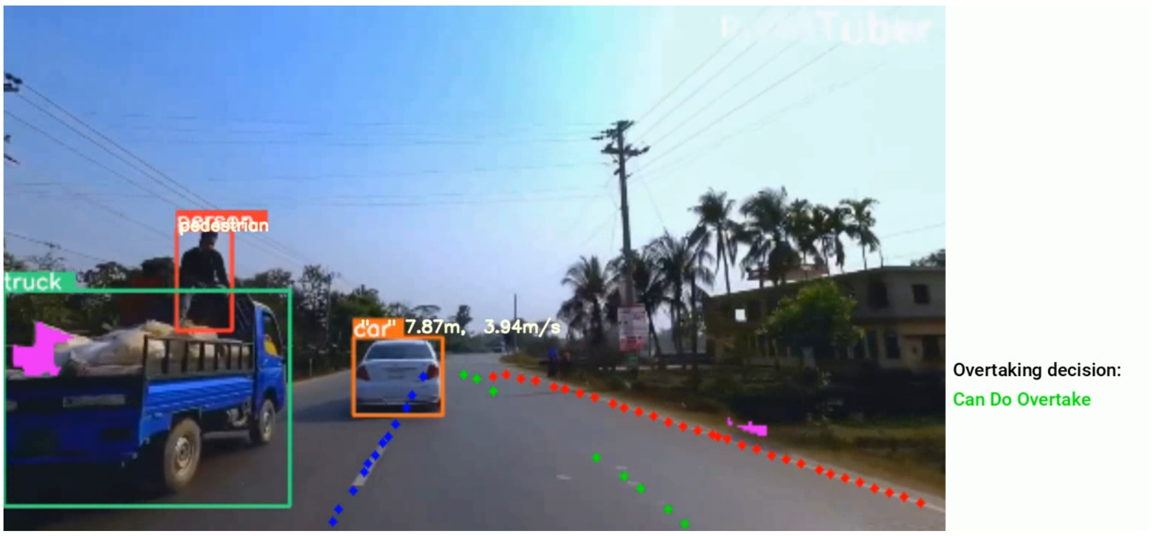

Figure 5.5: Overtaking Prediction

In figure 5.6, we can see that a vehicle is coming from the opposite direction, so it is not safe to take an attempt for overtake and model also predicts the same decision.

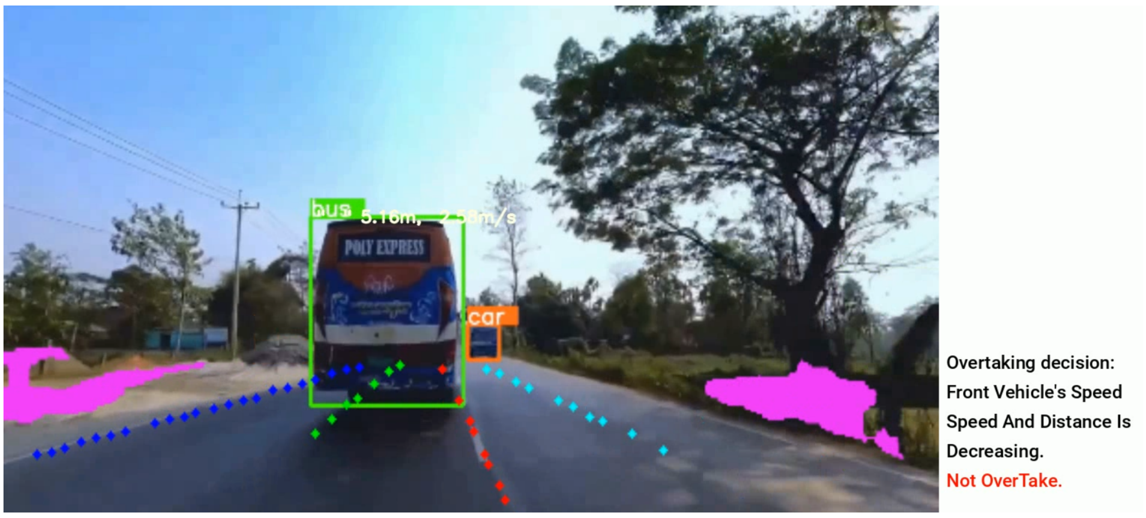

Figure 5.6: Not Overtaking Prediction

After that this figure 5.7 also shows almost the same scenario. Here, one vehicle is overtaking the bus already and also another vehicle is coming from the opposite side. So, it is also not a good decision to do overtake and our model goes with the same decision also.

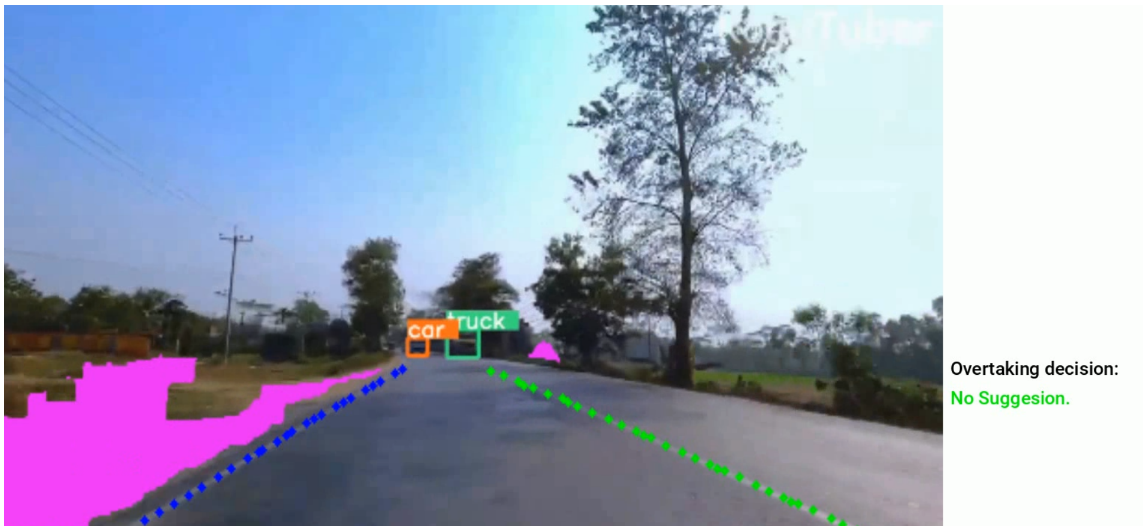

Figure 5.7: No Suggesion Prediction

For checking the accuracy of lane detection and velocity, we have faced several problems. As there is no labeled data set for velocity and lane detection based on our country. We couldn't check the accuracy rate. Besides, we have done one thing to check whether our model predicts the increasing or decreasing. We checked the velocity from the host vehicle and at the same time we also checked the velocity from the front vehicle. We used a mobile app to check the velocity and after that we saw that when our model says speed is decreasing or increasing then the speed is actually increasing or decreasing. And we have found that our model predicted the actual one regarding choosing a decreasing or increasing decision. In this way, we have checked the accuracy manually for speed. Moreover, for lane detection, as we mentioned earlier that there is no labeled data set for lane detection in our country, we have followed another way. We are fortunate that we have a car and a driver. We just sat behind the car and told the driver to see whether our car was in the right lane or not and after that we matched the value with the model prediction. After checking manually, we found that there were some noises in each frame as there was dirt, white paper etc. on the road.

Chapter 6

Conclusion

In conclusion, Road accidents are causing more unfortunate deaths than any other parameters in Bangladesh. Similarly, overtaking and over speeding accidents caused on highway roads are typical incidents in Bangladesh. There are many solutions out there but not a perfect solution considering Bangladesh's per capita, meteorological conditions etc. But our model is a promising technology that has the potential to significantly reduce the number of accidents caused by human error during overtaking maneuvers. It can detect objects and assess the risk of overtaking, and then allow the system to make safe and efficient passing of vehicles. Additionally, it can assist drivers in executing overtaking maneuvers by providing real-time feedback and control over the vehicle.

Besides, we intend to increase the accuracy and offer a perfect system. For this we make videos and apply YOLOv5 to detect vehicles by making dataset of Bangladeshi vehicles using Roboflow. By training and testing the dataset and merging all the measurement of speed and distance, the accuracy rate of our system is quite satisfactory. As the system is trained for Bangladeshi vehicles only, following this decision, a vehicle driver can play an important role in removing unexpected road accidents in our country.

However, additional study and development are required to enhance the system's functionality, guarantee safe and moral use, and resolve the difficulties and constraints related to its application. Future updates to the system could include the lane checking overtaking laws and regulations. Additionally, the overriding decision criteria may be more sophisticated for better suggesting accuracy. In future, it is possible to create a video with better camera so that we can get a better quality video input to have a more accurate result for this system. Then we can apply YOLOv7 to have the fine result in vehicle and all obstacle detection. Finally, occluded object detection can be added to the system to make it easier to identify the vehicles.

So, the effectiveness of our model with deep learning approach has been demonstrated through various studies i.e image processing, video processing, training and testing, thus we can strongly say that our model will contribute to our country and in the long run it will be established as a profound system for saving lives.

Bibliography

[1] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI’81, Vancouver, BC, Canada: Morgan Kaufmann Publishers Inc., 1981, pp. 674–679.

[2] J. Shi and Tomasi, “Good features to track,” in *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600. DOI: 10.1109/CVPR.1994.323794.

[3] S. Kim, S.-Y. Oh, J. Kang, Y. Ryu, K. Kim, S.-C. Park, and K. Park, “Front and rear vehicle detection and tracking in the day and night times using vision and sonar sensor fusion,” in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 2173–2178. DOI: 10.1109/IROS.2005.1545321.

[4] D. Gerónimo, A. M. López, A. D. Sappa, and T. Graf, “Survey of pedestrian detection for advanced driver assistance systems,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1239–1258, 2010. DOI: 10.1109/TPAMI.2009.122.

[5] C. Zhang, L. Wang, and R. Yang, “Semantic segmentation of urban scenes using dense depth maps,” vol. 6314, Sep. 2010, pp. 708–721, ISBN: 978-3-642-15560-4. DOI: 10.1007/978-3-642-15561-1_51.

[6] R. Okuda, Y. Kajiwara, and K. Terashima, “A survey of technical trend of adas and autonomous driving,” in *Proceedings of Technical Program - 2014 International Symposium on VLSI Technology, Systems and Application (VLSI-TSA)*, 2014, pp. 1–4. DOI: 10.1109/VLSI-TSA.2014.6839646.

[7] *Brta analyst: Overtaking, overload, overspeed cause most accidents*, Last Accessed on 23 March, 2023, May 2015. [Online]. Available: https://archive.dhakatribune.com/uncategorized/2015/05/24/brta-analyst-overtaking-overload-overspeed-cause-most-accidents.

[8] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” Jun. 2015, pp. 3431–3440. DOI: 10.1109/CVPR.2015.7298965.

[9] H. Xu, Y. Gao, F. Yu, and T. Darrell, “End-to-end learning of driving models from large-scale video datasets,” Dec. 2016.

[10] B. Chen, C. Gong, and J. Yang, “Importance-aware semantic segmentation for autonomous vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, pp. 1–12, Mar. 2018. DOI: 10.1109/TITS.2018.2801309.

[11] A. Kundu, Y. Li, and J. Rehg, “3d-rcnn: Instance-level 3d object reconstruction via render-and-compare,” Jun. 2018, pp. 3559–3568. DOI: 10.1109/CVPR.2018.00375.

[12] C. Mo, Y. Li, and Z. Ling, “Simulation and analysis on overtaking safety assistance system based on vehicle-to-vehicle communication,” *Automotive Innovation*, vol. 1, Jun. 2018. DOI: 10.1007/s42154-018-0017-9.

[13] M. Yang, Y. Kun, C. Zhang, Z. Li, and K. Yang, “Denseaspp for semantic segmentation in street scenes,” Jul. 2018. DOI: 10.1109/CVPR.2018.00388.

[14] V. P. Abdul Rahim Praveen Kumar Mallik, “Fractal antenna design for overtaking on highways in 5g vehicular communication ad-hoc networks environment,” *International Journal of Engineering and Advanced Technology*, vol. 9, no. 1S6, 2019. DOI: 10.35940/ijeat.A1031.1291S619.

[15] J.-C. Chen, Y.-F. Chen, and C.-H. Chuang, “Overtaking vehicle detection based on deep learning and headlight recognition,” in *2019 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*, 2019, pp. 1–2. DOI: 10.1109/ICCE-TW46550.2019.8991755.

[16] Z. Chen, Q. Liu, and C. Lian, “Pointlanenet: Efficient end-to-end cnns for accurate real-time lane detection,” *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 2563–2568, 2019.

[17] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, “Robust lane detection from continuous driving scenes using deep neural networks,” *IEEE Transactions on Vehicular Technology*, vol. 69, pp. 41–54, 2019.

[18] M. Athree and A. Jayasiri, “Vision-based automatic warning system to prevent dangerous and illegal vehicle overtaking,” in *2020 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, 2020, pp. 25–30. DOI: 10.1109/SCSE49731.2020.9313006.

[19] Y. Ko, Y. Lee, S. Azam, F. Munir, M. Jeon, and W. Pedrycz, “Key points estimation and point instance segmentation approach for lane detection,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 8949–8958, 2020.

[20] H.-Y. Lin, J.-M. Dai, L.-T. Wu, and L.-Q. Chen, “A vision-based driver assistance system with forward collision and overtaking detection,” *Sensors*, vol. 20, no. 18, 2020. DOI: 10.3390/s20185139.

[21] S. Shaira T. and S. Nakib, “Poribohon-bd,” 2020. DOI: 10.17632/pwyyg8zmk5.2.

[22] D. S. R. Shrinath Oza, “Object detection using iot and machine learning to avoid accident and improve road safety,” *International Journal of Engineering and Technical Research V9(06)*, vol. 9, no. 6, 2020. DOI: 10.17577/IJERTV9IS060640.

[23] W. Wang, H. Lin, and J. Wang, “Cnn based lane detection with instance segmentation in edge-cloud computing,” *Journal of Cloud Computing*, vol. 9, May 2020. DOI: 10.1186/s13677-020-00172-z.

[24] *Bangladesh 106th among 183 countries for having most road accidents*, Last Accessed on 23 March, 2023, Nov. 2021. [Online]. Available: https://www.tbsnews.net/bangladesh/bangladesh-106th-among-183-countries-having-most-road-accidents-report-335299.

[25] J. Li, D. Zhang, Y. Ma, and Q. Liu, "Lane image detection based on convolution neural network multi-task learning," *Electronics*, vol. 10, no. 19, 2021, ISSN: 2079-9292. [Online]. Available: https://www.mdpi.com/2079-9292/10/19/2356.

[26] G. Mandal, D. Bhattacharya, and De, "Real time vision based overtaking assistance system for drivers at night on two-lane single carriageway," *Computación y Sistemas*, vol. 25, May 2021. DOI: 10.13053/cys-25-2-3783.

[27] S. K. Perepu and P. Prasanna Kumar, "Safe overtaking using image processing and deep learning techniques," in *2021 IEEE International Conference on Computing (ICOCO)*, 2021, pp. 55–60. DOI: 10.1109/ICOCO53166.2021.9673539.

[28] N. Rahaman, M. Biswas, S. Chaki, M. M. Hossain, S. Ahmed, and M. Biswas, "Lane detection for autonomous vehicle management: Pht approach," Dec. 2021, pp. 1–6. DOI: 10.1109/ICCIT54785.2021.9689883.

[29] *Road crashes in bangladesh claim 413 lives in november*, Last Accessed on 23 March, 2023, Dec. 2021. [Online]. Available: https://www.dhakatribune.com/bangladesh/2021/12/04/report-road-crashes-in-bangladesh-claim-413-lives-in-november.

[30] S. Cakir, M. Gauß, K. Häppeler, Y. Ounajjar, F. Heinle, and R. Marchthaler, *Semantic segmentation for autonomous driving: Model evaluation, dataset generation, perspective comparison, and real-time capability*, Jul. 2022. DOI: 10.48550/arXiv.2207.12939.

[31] G. Cha, H. Jang, and D. Wee, "Self-supervised depth estimation with isometric-self-sample-based learning," *ArXiv*, vol. abs/2205.10006, 2022.

[32] H. Florea and S. Nedevschi, "Survey on monocular depth estimation for unmanned aerial vehicles using deep learning," in *2022 IEEE 18th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2022, pp. 319–326. DOI: 10.1109/ICCP56966.2022.10053950.

[33] S. Lee, K. Han, S. Park, and X. Yang, "Vehicle distance estimation from a monocular camera for advanced driver assistance systems," *Symmetry*, vol. 14, p. 2657, Dec. 2022. DOI: 10.3390/sym14122657.

[34] W. Ma and S. Zhu, "A multifeature-assisted road and vehicle detection method based on monocular depth estimation and refined u-v disparity mapping," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 16 763–16 772, 2022. DOI: 10.1109/TITS.2022.3195297.

[35] F. Mumuni, A. Mumuni, and C. K. Amuzuvi, "Deep learning of monocular depth, optical flow and ego-motion with geometric guidance for uav navigation in dynamic environments," *Machine Learning with Applications*, vol. 10, p. 100 416, 2022, ISSN: 2666-8270. DOI: https://doi.org/10.1016/j.mlwa.2022.100416. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666827022000913.

[36] C. Qi, H. Lv, G. Mu, C. Song, F. Xiao, S. Song, N. Zhang, and D. Wang, “Vehicle depth estimation based on monocular camera in extreme weather,” in *2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI)*, 2022, pp. 1–6. DOI: 10.1109/CVCI56766.2022.9964883.

[37] M. Weber, R. Polley, and J. M. Zöllner, “Learning implicit depth information for monocular 3d object detection,” in *2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, 2022, pp. 1–7. DOI: 10.1109/ICECCME55909.2022.9988525.

[38] Y. Yamada, A. S. M. Bakibillah, K. Hashikura, M. A. S. Kamal, and K. Yamada, “Autonomous vehicle overtaking: Modeling and an optimal trajectory generation scheme,” *Sustainability*, vol. 14, no. 3, 2022. DOI: 10.3390/su14031807.

[39] V. Saravanarajan, R.-C. Chen, C.-H. Hsieh, and L.-S. Chen, “Improving semantic segmentation under hazy weather for autonomous vehicles using explainable artificial intelligence and adaptive dehazing approach,” *IEEE Access*, vol. PP, pp. 1–1, Jan. 2023. DOI: 10.1109/ACCESS.2023.3251728.