Lane Navigation through Lateral-Longitudinal Control and Traffic Detection for Autonomous Vehicles

by

Ahnaf Asif Khan (160021006)
Sharara Hossain (160021014)
Abrar Islam (160021017)

A Thesis Submitted to the Academic Faculty in Partial Fulfillment of the Requirements for the Degree of

BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
Gazipur, Bangladesh

February 2021

Lane Navigation through Lateral-Longitudinal Control and Traffic Detection for Autonomous Vehicles

Authors

Ahnaf Asif Khan **Sharara Hossain**

---------- ----------

Abrar Islam

Approved by:

Mr. Mirza Fuad Adnan

Supervisor and Assistant Professor

Department of Electrical and Electronic Engineering
Islamic University of Technology (IUT)
Boardbazar, Gazipur-1704.

Date: ……

Abstract

Autonomous vehicle research is currently one of the trendings fields. This technology has immense potential to revolutionize present socio-economic dynamics. We address the notion of self-driving cars deployment in the context of Bangladesh. This thesis addresses mainly the perception and planning parts of an autonomous vehicle. The detection and tracking of objects around an autonomous vehicle is essential to operate safely. This paper can be divided mainly into two parts. Firstly, it presents the use of CARLA simulator for the purpose of lane detection and navigation of an autonomous vehicle and secondly, it shows the use of YOLOv5 for object detection. Both the longitudinal and lateral controls for lane navigation are done with the help of the built-in map in CARLA. And for object detection part, at first we have used SRGAN algorithm to enhance the quality of the images. Then using those enhanced images, YOLOv5 was used in order to detect objects in those images. For further improvement, these two different parts can be integrated by feeding the outcome of YOLOv5 to the CARLA simulator so that the autonomous vehicle can detect objects and obstacles.

Table of Contents

List of Figures

List of Tables

Acknowledgements

First of all, thanks and praises to the Allah Almighty, for His blessings in our study, so that our thesis could be successfully completed. I would like to express my heartfelt and profound gratitude to our thesis supervisor Mr. Mirza Fuad Adnan for providing invaluable help, encouragement, and insightful advice during our research. . We were profoundly influenced by his innovativeness, sight, sincerity and inspiration. We are immensely thankful to our family for their devotion, prayers, care and sacrifices in education and preparation for our future. Finally, we thank all those who helped us in a direct or an indirect way for the completion of the thesis.

1. INTRODUCTION

Ever since the invention and mass production of automobiles in the early 20th century, cars have been a driving force for developed economies. They have ensured faster than ever commute and ease of traveling. Flexibility in transportation for the layman has had a far reaching impact on economies and infrastructure in these countries.

Over the years. cars have gotten upgrades thanks to emerging technologies, therefore making it more passenger and environment friendly. However, humanity is yet to achieve full autonomy in these motorized vehicles. This mode of driving is to be achieved in autonomous vehicles or self-driving cars.

Self-driving cars have been a point of interest dating as far back as the 1930s. The first semi-automated vehicle dates back to the 1960s. The DARPA Grand Challenge in the 2000s made significant contributions to autonomous vehicle research. Thanks to ongoing research, many renowned companies have been deploying vehicles with various degrees of autonomy especially since the last decade. A trend in shifting to higher degrees of automation has been observed among leading automotive companies in recent times.

Academic researchers have played and continue to play a major role in advancing the technology that makes autonomous driving and fully automated vehicles a reality, as they have in other scientific and technical breakthroughs, by contributing groundbreaking and creative concepts, proof-of-concept systems, and a constant pool of talented engineers and entrepreneurs. These contributions come from a variety of engineers. Multiple engineering and computational science disciplines have contributed to this work. Intelligent systems orientation and real-world experimental projects have found a special niche in advancing the area of autonomous driving, not just in academic research groups but also, more recently, in industry, due to the design of these vehicles.

A self-driving car uses sensor data to perceive its environment, localizes itself, and then plans and executes a trajectory. For a vehicle to achieve autonomy, it would have to perform many complex processes from different fields of technology correctly, leaving room for minute variation only. Technologies like Adaptive Cruise Control and Lane Keeping Assist System are already widely in use.

A self-driving car is often referred to as an ego. The ego communicates with other systems such as vehicles, traffic infrastructure, cloud systems etc. It is collectively known as V2X (vehicle to everything) communication. IoT technology is vital for

autonomous systems as it is based on sensor data and communication between such devices and systems.

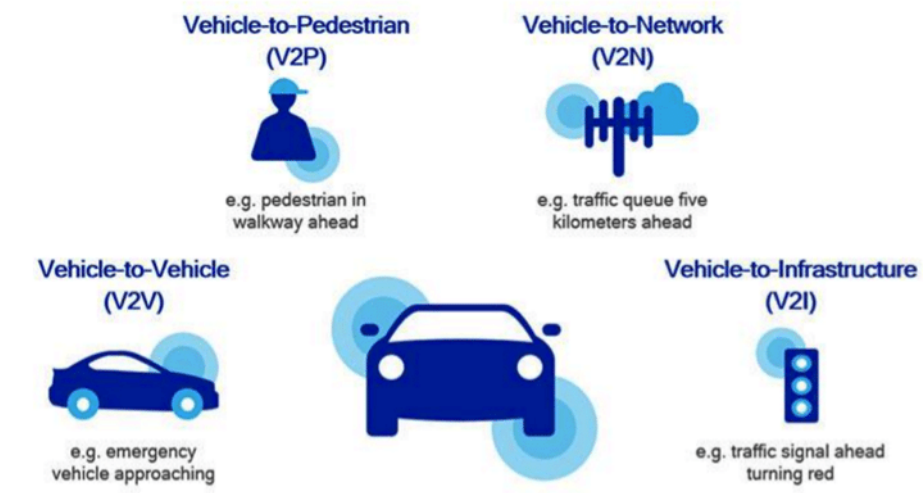

Figure 1: V2X Communication of self-driving car (Image Source: [13])

Among the many benefits of autonomous vehicles, safety and cost effectiveness are the principal ones. It can also contribute to reducing traffic congestion, which in turn can reduce carbon emission in the atmosphere. We can also count on increased efficiency in time and fuel consumption in transportation. Car sharing services may become more popular. Autonomous vehicles may promote healthy urban environments if deployed safely. They can also increase productivity and housing affordability [12]. Along with elderly and persons with disability, AV can increase mobility for the young and low-income groups. Higher speed limits and smoother rides can be achieved. We can expect great benefits in rescue and emergency response situations. Parking problems in big cities could be all but eliminated.

Apart from the many benefits there are some perceived risks to AV technology. For successful deployment, large investments and new road systems will have to be made. In fact, the entire transportation system, services and infrastructures will have to be rebuilt. There is also the potential of system failure, since the AV system depends on so many processes. Another issue to contend with is the cyber security and privacy concerns since the system will be connected to the cloud. There are also issues related to liability, global standards for implementation and the time it might take to replace current transports with autonomous vehicles. Ethical concerns have also been raised in the event of an unavoidable crash. Today's artificial intelligence has not achieved the robustness to work in a chaotic real-life environment.

Vehicles that are self-driving or highly automated must navigate smoothly and avoid obstacles while correctly comprehending the highly nuanced semantic meaning of the environment and dynamic activities in its field of vision. Further advancements in perception, 3D scene comprehension, and policy planning are needed to achieve such objectives

Autonomous driving can be broken down to three components - sensing, perception and control or decision. Sensing includes sensor feed and control boils down to decision and execution based on complex reasoning - both mostly hardware. Designing autonomous vehicles that are efficient, safe and useful is quite a daunting challenge. The design is complex and consists of many technological fields and requires a full synchronization in hardware and software integration. Alongside these, ODD (Operational Design Domain) is something that plays a pivotal role in the driving task. The driving task is made up of lateral and longitudinal control, object and event detection and response (OEDR), short term and long term planning, etc.

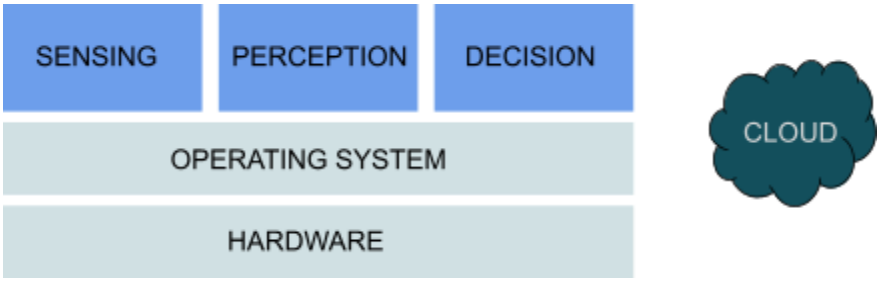

Figure 2: AV driving system overview

Our area of focus for this thesis is perception and decision. Perception entails localization, object recognition and object tracking. In simple terms, the task of perception is to localize the ego i.e the self-driving car, and to map the environment and understand it, then detecting objects, tracking patterns to make predictions. The perception goals are identifying static objects (like curbs and street signs), dynamic objects and their motions (like other vehicles and pedestrians on the street), ego localization (where we are and how we are moving) – position, velocity, acceleration, orientation, angular motion etc.

The algorithm subsystem utilizes raw data from the sensors to understand its environment and to make decisions about its future actions. Then there is the behavior decision module which can be considered as a “co-driver” that takes decisions about

navigation. It leverages insights generated from the perception module to make effective and safe decisions.

Self driving cars can have five levels of autonomy. Levels 1 through 5. Level 1 offers either lateral or longitudinal control. Level 2 offers both. Level 3 includes automated object and event detection and response. Level 4 can handle emergency situations autonomously. Level 5 is a fully autonomous vehicle.

In this research, we implement lateral and longitudinal control of a self-driving car using an open-source simulator. For perception, we selected the context of Bangladesh as it presents a challenging task. We employed an artificial intelligence technique deep learning to perform object detection. The dataset [15] has been collected from an online competition Dhaka AI Traffic Detection challenge 2020. In the future, we intend to merge these two tasks in order to improve the level of autonomy of our designed system.

1.1 Purpose and Research Question

Research shows that 95% of vehicle crashes are due to human error [1]. Autonomous vehicle research has the prospect of reducing the percentage significantly saving both lives and money. Another scope for autonomous vehicles is that it can provide mobility to the ageing population and people with disabilities. Another important motivation is that autonomous vehicles can reduce the number of vehicles that sit idle for most of the day. Building an affordable self-driving car is a matter of this time. Research into self-driving cars can have business potential as the trends are leading that way.

We particularly focus in the context of Bangladesh where traffic is congested and hundreds of thousands of productive hours are wasted in traffic jams. According to BUET accident research centre, the fatalities due to road accidents in Bangladesh reach around 10 thousand every year, while many more end up severely injured and disabled. Autonomous vehicles can bring about revolutionary changes in the transportation sector in Bangladesh.

As a first step, we start with perception. Using the Dhaka AI dataset, we set to answer the question of whether traffic detection is possible considering the number of types of vehicles that roam the roads of Dhaka. Then we move on to simulating a self-driving car in a virtual platform CARLA.

Research question: Is it possible to design a low-level self-driving car with such perception capabilities that can detect the various types of vehicles on the streets of Dhaka?

1.2 Approach and Methodology

We used the CARLA simulator to emulate an ego's performance for steering and planning in a controlled environment. It is an open-source simulator for autonomous vehicle research. CARLA supports versatile specifications of sensors, environmental conditions, complete control, maps generation, etc. We have also used a deep learning model made of YOLO version 5 and SRGAN for object detection in still images. YOLOv5 is a fast object detection architecture that uses grids in an image unlike CNN networks. SRGAN is used to enhance the image quality for better training and detection. The dataset used is a public one that contains street images of different locations in Dhaka, Bangladesh.

We focus on the object detection task of perception and steering task of planning in self-driving cars.

1.3 Related Work

In this part we do an abridged literature review based on relevant work. This will help us gain a better understanding of the kind of work that's being done i.e. the context.

Daily et al. explores technological advances in self-driving cars and its future [16]. Self-driving technology has the potential to markedly impact the global transportation scenario. Globally 10 trillion miles are driven each year, with unique and complex situations brought by each vehicle. This poses one of the principal challenges for autonomous vehicle researchers.

For end to end learning for autonomous vehicles, Bojarski et al. trained a CNN (Convolutional Neural Network) architecture to map image data to steering commands [17]. This system offers a small network which is valuable considering the limited computational power available to a self-driving vehicle.

A direct perception approach, DeepDriving proposed an approach to map small key indicators in perception. A deep Convolutional Neural Network was trained using recordings from 12 hours of human driving in a video game. The model works well in a very diverse set of virtual environments. This refers to a direct perception approach to estimate the affordance for driving [2].

In [7], Lee et al. explore and modify pre-trained CNNs (Convolutional Neural Networks) AlexNet and GoogLeNet that map an input image to few perception indicators for calculating driving affordances in highway traffic. They also designed a controller with these indicators along with the short-range sensor information from TORCS (open racing car simulator) for driving simulated cars to avoid collisions on the road.

One of the major open challenges in self-driving cars is the detection of other vehicles on the street and pedestrians for safe navigation. Srinivasan et al. proposed an automated method to identify mistakes made by object detectors without ground truth labels. They showed that inconsistencies in object detector output between a pair of similar images can be used as hypotheses for false negatives. They employed a novel set of features for each hypothesis. With this, a binary classifier was used to find valid errors [3].

Cameras are a crucial exteroceptive sensor for self-driving cars. Exteroceptive sensors record data from the ego's surroundings. Cameras are cheap and small, they provide information about the environment similar to the human eye, and work in various weather conditions, unlike many exteroceptive sensors. In [4], Häne et al. describe the camera calibration and subsequent processing pipeline for multi-fisheye-camera systems developed as part of the V-Charge project which eliminates the need for use of multiple cameras. This project aims to enable automated valet parking.

The perception subsystem utilizes onboard sensors like cameras and LiDARs (Light Detection and Ranging) to assess the ego's surroundings. In [5], Sun et al. perform a study to explore vulnerabilities in the current LiDAR-based perception architectures and discover that the ignored occlusion patterns in LiDAR point clouds put self-driving cars at risk of spoofing attacks. A model-agnostic defense solution CARLO is proposed to mitigate LiDAR spoofing attacks.

Object detection is one the core tasks of perception [21]. In [6], Philion et al. propose a metric for 3D object detection particularly for the task of self-driving. The mainconcept of this metric is to isolate the task of detection and measure the impact the produced detections might induce on the task of driving.

CARLA is an open source urban driving simulator. Dostoyvitsky et al. built this simulator to observe and study the performance of autonomous driving in three approaches: a classic modular pipeline, an end- to-end model trained via imitation learning, and an end-to-end model trained via reinforcement learning [8]. These are evaluated in controlled scenarios of increasing difficulty, and their performance is evaluated using metrics provided by CARLA. This platform has immense contribution to autonomous driving research.

A vehicle trained end-to-end to imitate an expert cannot be guided to take intelligent decisions instantly. This is a limiting approach. Codevilla et al. propose to condition imitation learning on high-level command input and evaluate different types of architectures in vision-based driving [9].

An autonomous vehicle has to deal with real life situations that can vary significantly from the test data it has been trained on. In this case, GAN (Generative Adversarial Network) architectures can be used to augment realistic data for training. An empirical analysis followed by validation allowed Xu et al. to propose a solution based on GAN based data generation for the ego's perception module [20].

In [10], SRGAN learns how to restore realistic high-resolution images from down-sampled ones; they propose two approaches. The first one is a down-sampling method using noise injection to create desirable low-resolution images from high-resolution ones for model training. The second one is to train models for each target domain: they use the beef, bread, chicken and pound cake categories in their experiments. They also propose a novel evaluation method, Xception score. Compared with existing methods using qualitative and quantitative experiments, we find the proposed methods can generate more realistic super-resolved images.

Boss, an autonomous vehicle won the 2007 DARPA Urban Challenge [18]. Boss can park, pass static and dynamic objects beyond basic driving. The Boss system consists of perception, motion planning, mission planning, infrastructure and behavioral planning.

A visual ego-motion algorithm has been developed for self-driving cars with multi-camera systems by Lee et al [19]. Generalizing the camera vision, full motion including metric scale has been obtained. They have also shown that the results hold for real-life data as well.

1.4 Scope and Limitations

Deploying autonomous vehicles in Dhaka would bring its own unique challenges. One of the main reasons for heavy traffic in Dhaka, is due to the increasing number of private cars. Ride-share services like Uber, Pathao are transforming Dhaka's transport sector. According to a report by Uber, a private car remains unutilized 96% of the time. Also it is costly owning and maintaining a car which is only utilized 4%. There are also vehicles like rikshaw and CNG which are not common in other countries. So in order to test a self-driving car in Dhaka streets, we developed a model trained on a dataset based on this city.

Among the major challenges in designing autonomous vehicles remain in software integration (due to the number of processes, sensors and safety issues), failure in prediction (deep learning models are yet to achieve the level of accuracy needed) and scaling. So we focus on very specific tasks instead of going all in.

We have used the CARLA simulator to test our approach. But there are some limitations. In Bangladesh, due to heavy traffic there is a presence of a lot of vehicles at a time around the user. That much traffic can not be simulated in the CARLA environment. As it is mentioned earlier, if we summarize our work, there are mainly two parts - a) Lane detection and navigation and b) object detection. We have done the first part in CARLA using an HD map and PID and the latter part using YOLOv5. For future work, we will integrate these two parts, meaning the output of YOLOv5, i.e.images with detected objects which can be fed to serve the purpose of navigation.

We have achieved level 2 autonomy by CARLA simulation. By integrating object detection we can move to level 3 autonomy. The levels of autonomy are explained in section 2.2.

2. THEORETICAL BACKGROUND

2.1 Autonomous Vehicles

2.1.1 Definition

Autonomy

To learn about the autonomous vehicle, we need to learn about the meaning of autonomy first. Autonomy is the potential or a state of being self-governing; behaving independently from others. If we apply this concept on cars, those are described as the vehicles that are able to function without any human intervention, by using artificial intelligence.

Autonomous Vehicle

An autonomous vehicle can operate itself without human intervention by sensing its surroundings and carrying out the necessary functioning. In order to adapt to external conditions that a human driver handles, the automotive vehicle utilizes a fully automated driving system. It uses different in-vehicle techniques, such as adaptive cruise controls, active guidance or wire guidance, wire-break, GPS for navigation, lasers, radar and numerous sensors from beginning to predefined destination.

2.1.2 Autonomous Driving System Architecture

Autonomous driving is not only one technology, but a dynamic structure made up of several sub-systems. The framework can be divided into three key elements [22] :

- · Algorithm components
- · the Operating System
- · the Hardware System

In order to learn about its surroundings and make decisions about its future behavior, the algorithm subsystem derives valuable information from raw sensor data. These algorithms are incorporated by consumer applications in order to satisfy criteria in real time and reliability.

2.1.3 Algorithm Components

The components of the algorithm is as follows - the first one is sensing, which collects useful data from raw sensor data; the second one is perception, which is to spot the vehicle and evaluate the surrounding environment; and the third one is the decision, which means intentions to make sure that intended destinations are managed to reach safely. Following is a brief discussion on the mentioned components –

Sensing: For the sensing part, an autonomous vehicle uses several sensors. As any sensor category offers advantages and disadvantages, multiple sensor data for improved efficiency and protection must be integrated in standalone automobiles [23]. Following are some examples of most commonly used sensors –

· LIDAR: LiDAR is like a vehicle's eye. It gives them a 360° range of view that helps them drive safely. It is used to map, locate and prevent obstacles. This ensures the creation of a 3D-point cloud with light reflections, which is great for assessing scene geometry. It shoots light beams at the environment and measures the reflected return. An on-board computer handles the reflecting point of every laser and converts it into an animated three-dimensional depiction [24]. It is not affected by environment light, unlike a camera vision.

· GPS/IMU: These are proprioceptive sensors. GPS uses satellites to provide the vehicle navigation system with location information. The GPS receiver can measure its location within a few meters by obtaining a few tens of global satellites optimally. The GPS receiver cannot provide really precise location data and its update frequency is also sluggish. It is IMU's (Inertial Measurement Units) job to measure the location of the vehicle between any GPS receiver update. It measures angular rotation. Updates at or

above 200 Hz can be given by the IMU more regularly. We can provide detailed and real-time vehicle position alerts by way of integrating both GPS and IMU [26]. GNSS measures vehicular position and velocity. Wheel odometry is also used as a proprioceptive sensor.

· Cameras: In addition to sensor technologies such as LiDAR, RADAR, and ultrasound, 3D cameras are also used in order to ensure the correct synchronization of motions, allowing the autonomous vehicle to reliably identify its location and that of its targets at all times. It is essential for correctly perceiving the environment. However, a tradeoff is often made between resolution and field of view due to limited computational resources [28]

· RADAR: The RADAR is a strong additional sensor that can overcome the shortcomings of other sensors. RADARs can explicitly calculate the speed of an object and can run during multiple situations, e.g. day, night, fog, storms, snow and adverse weather in general. It can detect large objects robustly.

· SONAR: Sonars are used for short-range-all-weathers distance measurement. They are ideal for low cost parking protocols. They are also unaffected by lighting and precipitation [30].

Perception: The perception of autonomous vehicles is the system that perceives the world, i.e. the mechanism for the identification of data from sensors. Perception refers to the capability of an individual method to gather information and derive environmental related knowledge. Localization, object detection and object monitoring are the three primary features of autonomous driving perception. The sensing process in self-driving cars utilizes a mix of high-tech sensors (mentioned above) and cameras together with sophisticated algorithms to process and understand the vehicle's surroundings in real time. OEDR is one of the main tasks of the perception subsystem. It makes sense for the environment, performs identification and understands patterns to make predictions.

The environment perception modules have two key responsibilities. First, identifying the current location of the ego in space. Also, classifying and locating important elements of the environment necessary for the driving task [31]. Examples of these elements include other vehicles, pedestrians, the road and its markings, road signs i.e. anything that directly affects the task of driving.

The environment mapping module generates a series of maps that identify objects in the environment surrounding the autonomous vehicle for a variety of purposes, including collision avoidance, ego-motion monitoring, and motion planning. What steps to take and where to travel are all part of the motion planning process. The direction is taken by the controller, who adjusts the brakes and other gear positions. The system supervisor also alerts of any safety issues and failures.

Some of the challenges faced by the perception subsystem are - detection and segmentation, access to large datasets, sensor uncertainty, occlusion, reflection, illumination and lens flare cause sensor data loss, weather and precipitation.

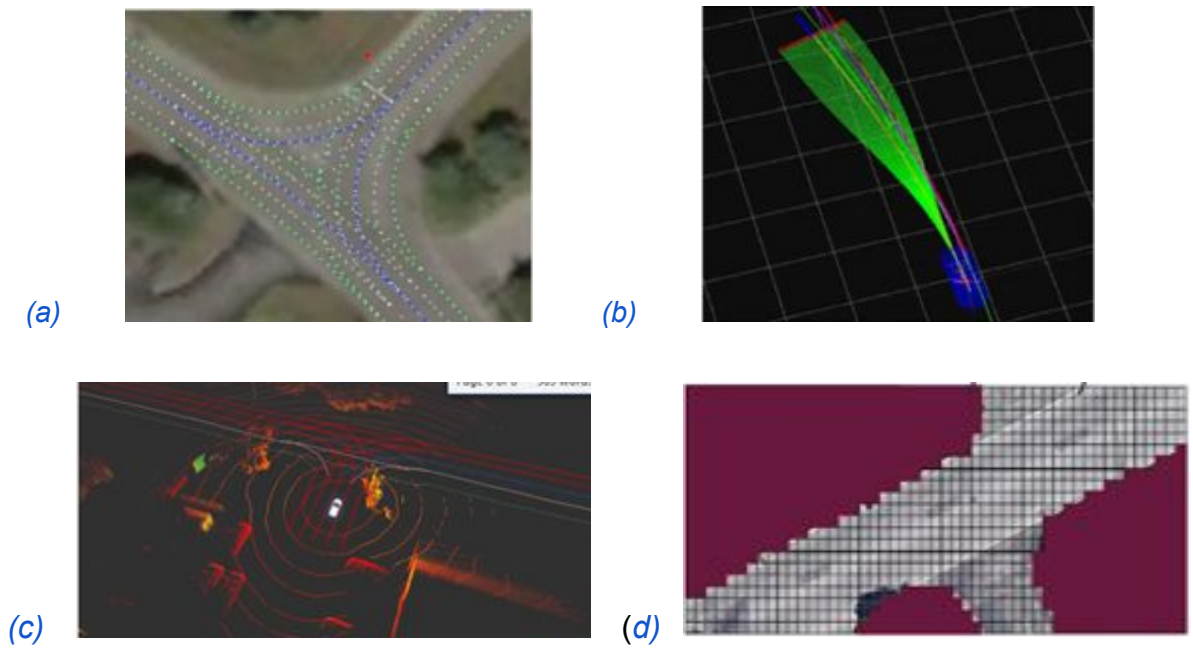

Figure 3: Environment mapping (a) Detailed road map, (b) Planned Trajectory, (c) Lccaliation map, (d) Occupancy grid (Image Source: [11])

Decision: And the last part of the algorithm components is decision. With the help of the sensors autonomous cars see the world. After perceiving the world, through the optimal algorithm, the car makes its own decision, which action to take. Normally, Deep Neural Network (DNN) is used for object detection and reinforcement learning is used for navigation. Based on object detection results and the learning curve from reinforcement learning, an autonomous car makes its decision [32]. Planning can be rule-based or predictive.

2.1.4 The Operating System

The systems incorporate the algorithms above in order to satisfy the criteria of real time and dependability. There are some challenges in this regard – if one of the devices fails, it must be sufficiently durable to rebound from the malfunction, as well as complete computation under strict energizing and capacity allocation. The pipeline must be processed rapidly enough to process the large amount of sensor data produced. Both serial and parallel processing are done, especially for image and LiDAR data processing.

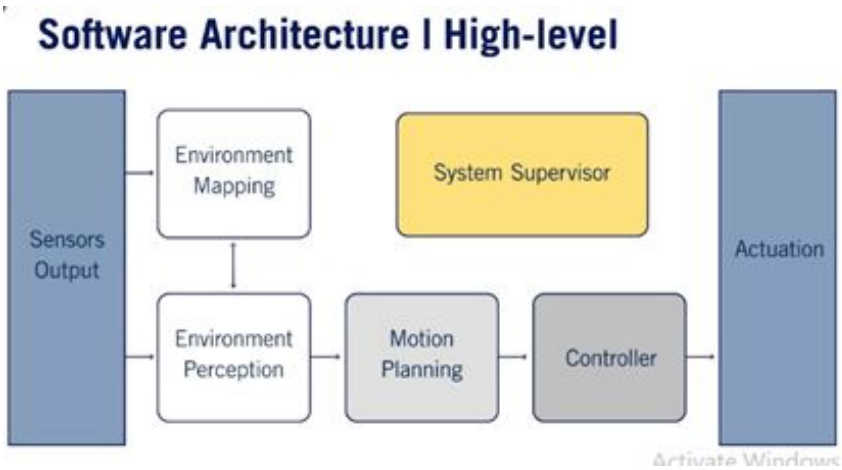

Figure 4: High Level Software Architecture for AV (Image Source: [11])

2.1.5 The Hardware System

For instance, we can mention the computing platform implementation from a leading autonomous driving company. It contains two device boxes, each with an Intel Xeon E5 processor and four to eight GPU accelerators from Nvidia K80. The second measurement box does almost the same functions and is used to ensure the reliability of the second box.

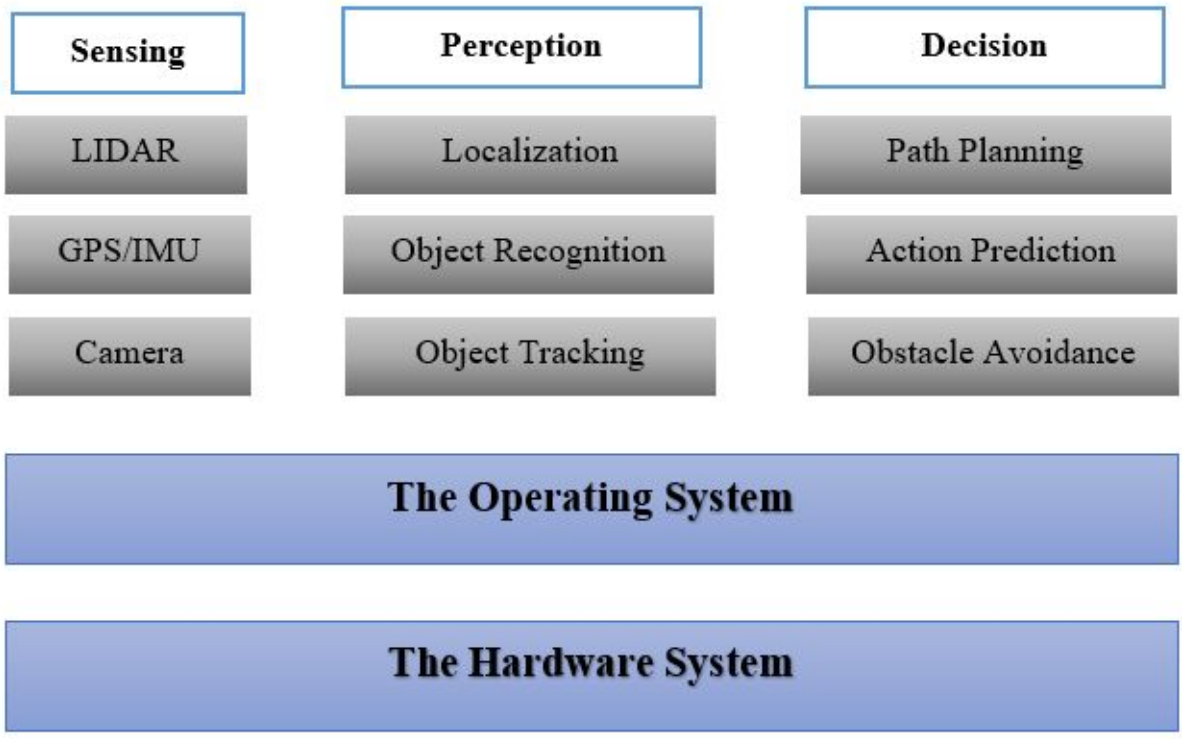

Figure 5: Architecture of autonomous vehicle system

2.2 Stages of Autonomous Vehicle

It is not easy to decide whether a car is self-driving or not. Instead, autonomous vehicles are accessible on several levels that allow one to know how individuals and vehicles communicate with a car pilot. The five automation stages are –

- Level One – Driver Assistance
- Level Two – Partial Automation
- Level Three – Conditional Automation
- Level Four – High Automation
- Level Five – Complete Automation or Full Autonomy

There is also a level called 'Level - Zero' referring to no automation mode. The driver carries out all the work activities, such as steering, braking, accelerating or decelerating, etc [33]. These stages demonstrate how the vehicle can behave and respond on its own. Following is a clear description of what each stage of automation should expect.

2.2.1 Level One – Driver Assistance

The vehicle can assist some tasks at this stage, but the driver mostly takes control of the acceleration, braking and tracking of the surrounding environment. In this stage the automated car can either take the longitudinal control or the lateral control, not both at a time. Nowadays, there are scarcely any modern vehicles lacking assistance systems; the bulk of cars are fitted with support functions of level 1. For example, if a car is too close to another car, the first stage of a vehicle will give a brake assist, or the adaptive cruise control feature can regulate the distance and speed, it could have a parking aid feature, where the driver can be alerted by a sound alarm and also it could provide lane departure warning [34]. The Nissan Sentra with its Smart Cruise Control feature of 2018 will be a typical example of Level One stage vehicle.

Figure 6: Level 1 AV (Image Source: [11])

2.2.2 Level Two – Partial Automation

Level 2 goes a step further. Several support systems here are mostly merged to allow the vehicle to execute individual driving maneuvers independently. In this stage the automated car can both take the longitudinal control or the lateral control at a time. This stage of cars can control both steering and braking/acceleration. The human driver must still take care of the whole challenge and conduct the remainder of the driving task which is referred to as "driving environment monitoring". Level 2 vehicles support the centering of the car within the lane in a steering mechanism, while the speed control feature guarantees the right distance from other vehicles is retained. Current vehicle models are often classified as level 2 and while the functions are often spectacular, it is important to point out that they are in no way true self-driving functions, which cause drivers to distract from the road. Driver's attention is always needed. The Volvo S60 2019 with its self-braking function and pilot assistance capability is a clear example of a Level 2 autonomy vehicle.

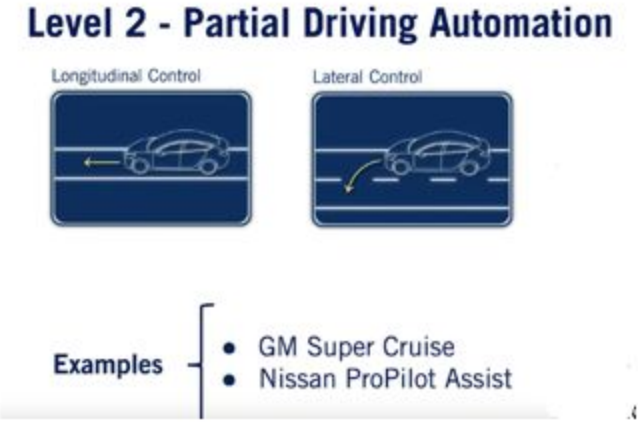

Figure 7: Level 2 AV (Image Source: [11])

2.2.3 Level Three – Conditional Automation

The technical disparity between level 2 and level 3 is large. Although Level 2 is all about support structures, level 3 simply means that under some circumstances the car drives autonomously. The environmental monitoring is managed by the vehicle with sensors such as LiDAR. Like level -2, this stage also has the longitudinal and lateral control. Moreover, this level provides the car with automated object and event detection and response (OEDR). This level is also called eye-off cars, which allow drivers to concentrate on other acts such as using a cell phone. Particularly noticeable is the added advantage of automating such situations of highway driving: straight forward driving for long stretches is dull and comparatively simple compared with navigating urban road traffic. Driver attention isn't always necessary. In situations that are near the highway road exit or more complicated, such as road work, the driver takes over the control. On public roads that are not restricted access roads, there are not many level three automobiles. That being said, Honda will be launching a level three car soon.

Figure 8: Level 3 AV (Image Source: [11])

2.2.4 Level Four – High Automation

At level 4, the autonomous vehicles mechanism must first alert the driver of the safety of the conditions and only then would the driver turn the car into that mode. The next stages after the preliminary automation (Level 1-3) are to automate all driving conditions, i.e., fully autonomous Level 4 driving, where the car manages all journeys on the road and on urban traffic largely independently. This level of automation is appropriate for the cities' traffic condition, here in Bangladesh. Along with longitudinal, lateral control and OEDR, this level of vehicle allows fallbacks and alerts the driver to control the vehicle during emergency situations. Google's Waymo Project in the U.S. is the best instance of an independent vehicle level 4.

Figure 9: Level 4 AV (Image Source: [11])

2.2.5 Level Five – Complete Automation

This level of autonomous should not require human interaction. The control over gear, braking or a steering wheel are not needed, as the autonomous car system controls every critical task, environmental detection and recognition of specific driving situations such as traffic jams. Perhaps only in theory, Level 5 autonomy allows a vehicle to drive anywhere without a driver in all circumstances. The driver becomes a pure passenger. The user just specifies the location and endpoint. This leaves open totally new possibilities for cost-effective person transportation, since completely automated robo-taxi fleets can be run very economically. This level has all the features that level – 4 has. Additionally, this level provides unlimited Operational Design Domain (ODD). Although we still do not see a vehicle achieve true autonomy at level 5, the concept Analysis of Audi AI:CON gives an insight into how a completely autonomous car might look [35].

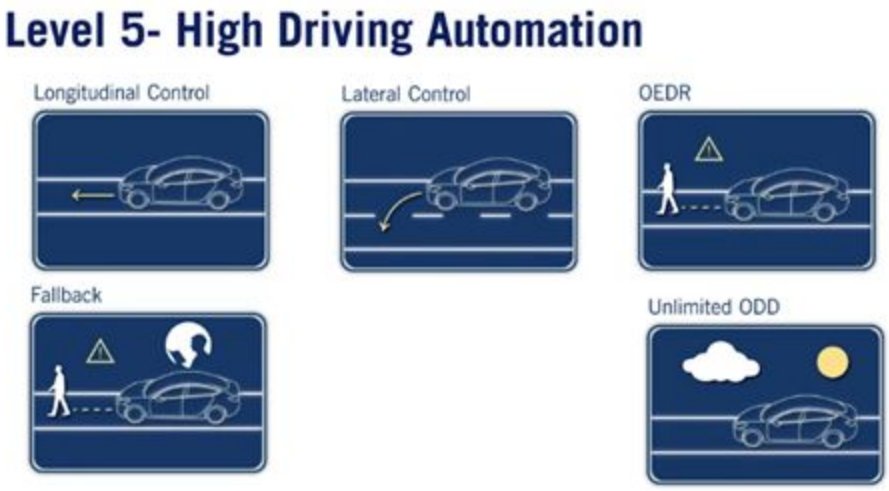

Figure : Level 5 AV (Image Source: [11])

Level No.	Level Name	Characteristics	Example
Level 1	Driver Assistance	Either longitudinal control or lateral control	The Nissan Sentra (2018)
Level 2	Partial Automation	Capable of both longitudinal control and lateral control	The Volvo S60 (2019)
Level 3	Conditional Automation	Alongside longitudinal and lateral control, presence of OEDR	Honda's Legend (2021 expected)
Level 4	High Automation	Capable of handling emergencies on its own alongside longitudinal and lateral control and OEDR.	Google's Waymo Project (2011-ongoing)
Level 5	Complete Automation	Fully automated. Has unlimited ODD.	Audi AI:CON (Concept)

Table 1: Stages of autonomous vehicles at a glance

2.3 Object detection

To grasp the picture better, classes of object instances must be defined as in object classification. In addition, they must also be spotted in orbit. This process is generally called object detection. The object detection subtask itself is one of the most critical pre-conditions for autonomous control in various autonomous vehicles, since this role enables the automobile controller to consider obstacles while considering future patterns. Object recognition systems take the input of images which typically output bounding boxes where labels of classes are mentioned for all objects of interest. As it is very important for the proper navigation of the autonomous vehicle, the algorithms for the object detection need to be as accurate as possible. In recent times, many high-quality object detectors, mainly convolutional neural network (CNN) models have undergone impressive developments. A Convolutional Neural Network is a Deep Learning algorithm. It takes an input and assigns weights and biases from various aspects /features of the image, which can be called "importance" of that particular aspect /feature. The model learns to discern one image from another through these features. Some mentionable models are –

- · OverFeat
- · VGG16
- · Fast R-CNN
- · YOLO
- · SimpleNet
- · SSD

Among these models, we will be discussing YOLO as we have used the YOLOv5 model for object detection.

2.3.1 YOLOv5

Following the resounding success of the first three versions of YOLO (You Only Look Once), Ultralytics launched YOLOv5 recently. YOLO is a Darknet based architecture that employs CNN for object detection in image and video format data. Due to its faster speed in training and inference compared to other object detection networks, it is commonly used for real-time applications which makes it ideal for autonomous vehicle perception.

Instead of using sliding boxes like a typical convolutional network, it divides the image into sxs grids. YOLO uses bounding boxes to identify and classify objects which allows it to detect multiple objects in a single image.

Figure 10. Object detection using YOLOv5 (Image Source: [14])

In a single forward pass, YOLO detects probabilities and bounding boxes for each object which gives it the name You Only Look Once. The object detection task is not done sequentially, rather by use of convolution. For each detection, a label vector containing coordinates of the bounding box, class and its confidence score is generated.

Anchor boxes are pre-made boxes with a specific height and width. The idea is to use a limited number of anchor boxes so that any object detected can fit snugly inside at least one of them. This is done to reduce the number of potential height and width values to only a few predefined options.The application determines which anchor boxes are used. The Main concept is that different objects have different height and width ratios. Use of anchor boxes reduces computational cost.

The model may yield multiple bounding boxes for the same object, Non-max suppression is used to discard the irrelevant boxes and keeps the box with the highest probability. It uses the concept of IoU (Intersection over union).

Version	Release (year)	Major Improvements
YOLOv1	2016	Taking as a regression problem, detecting objects in spatially segregated bounding boxes and related class probabilities.
YOLOv2	2017	Faster RCNN, 40-90 frame per second (FPS)
YOLOv3	2018	Helping simply to balance speed and precision by adjusting the scale of the model without training again.
YOLOv4	2020 (April)	Improve detector accuracy with same inference time.
YOLOv5	2020 (June)	Mosaic data augmentation and auto learning bounding box anchors.

Table 2: YOLO versions and improvements

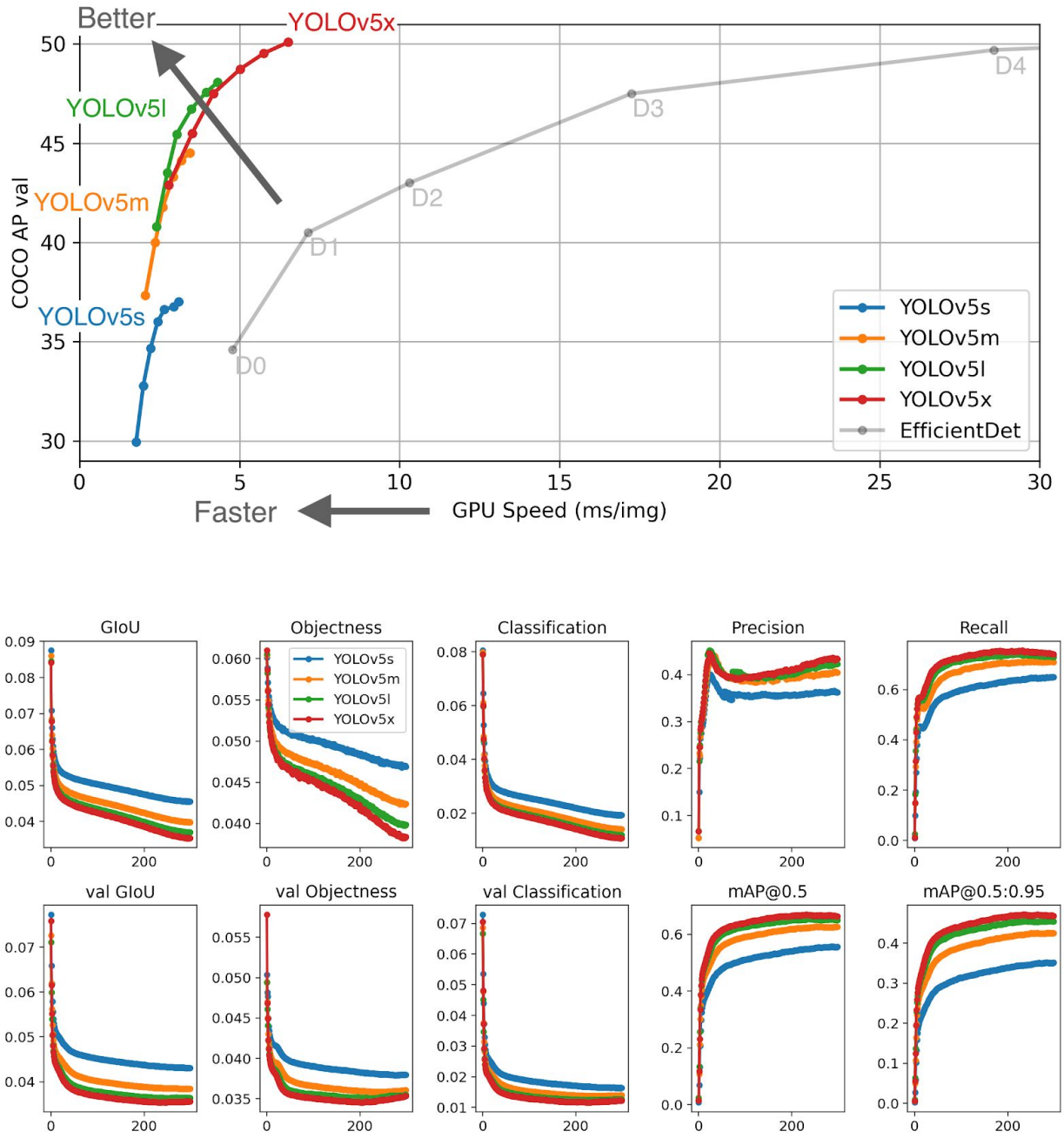

Figure 11: YOLO versions comparison (Image Source: [14])

As it is apparent from figure 11, YOLOv5 outperforms its previous versions.

3. METHODOLOGY

The whole thesis work is divided into two main parts.

- Lane detection and navigation using CARLA and
- Object detection using YOLOv5

Figure 12 depicts the system overview

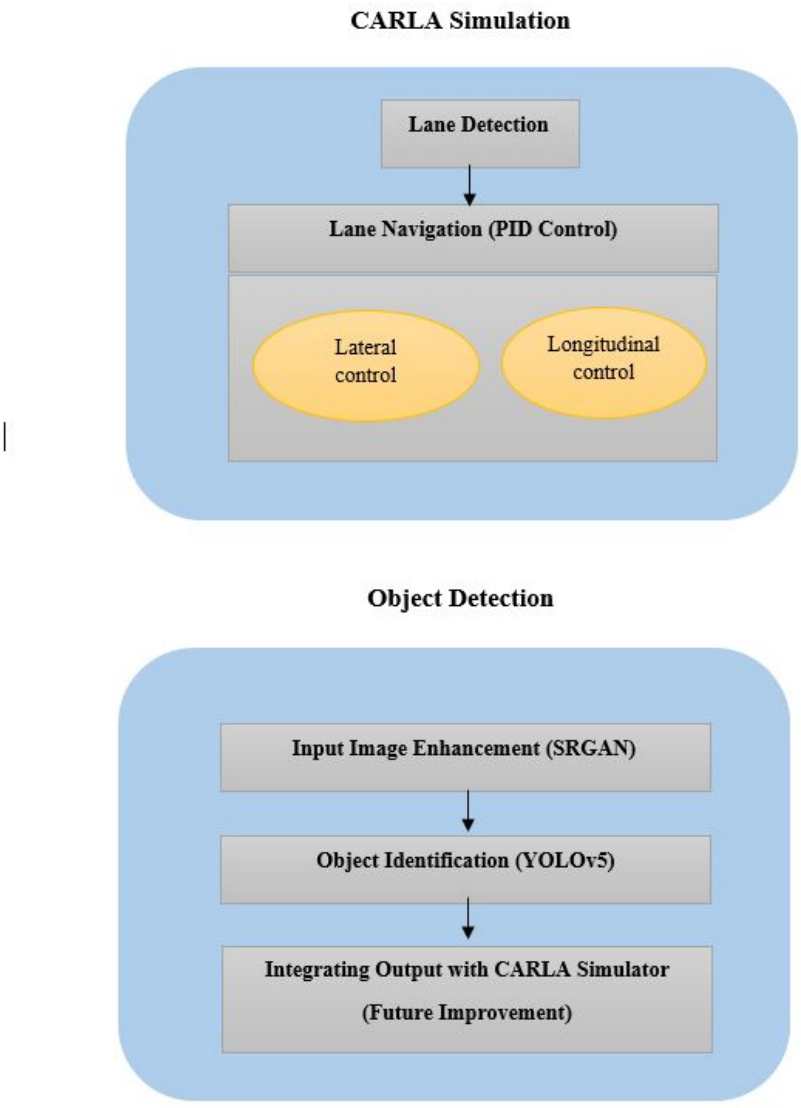

Figure 12: Overview of the system

3.1 Dataset

The dataset [15] is made up of vehicle images in the streets of Dhaka, with each image containing a vehicle from one of 21 different vehicle classes. This dataset is useful for multiple vehicle detection and recognition. The classes are: ambulance, auto-rickshaw, bicycle, bus, car, garbage van, human hauler, minibus, minivan, motorbike, Pickup, army vehicle, police car, rickshaw, scooter, Suv, taxi, three-wheelers (CNG), truck, van, wheelbarrow. This dataset contains images in various lighting situations including day and night.

3.2 Use of CARLA Simulator:

CARLA (Car Learning to Act) is an open source simulator for urban driving [8], which is implemented over Unreal Engine 4 (UE4). CARLA provides us with the environment and sensor data, which facilitates the simulation of our self-driving car.

Environment: The environment consists of both static and dynamic objects, where buildings, traffic signs and infrastructures are static, and vehicles and pedestrians are dynamic. The vehicles and sensors can be spawned at any specific location as per need.

Figure 13: Visualization of custom CARLA environment

Sensor: Sensors include a number of image sensors, RADAR, LIDAR and pseudo-sensors which provide us with the ground truth depth and semantic segmentation.

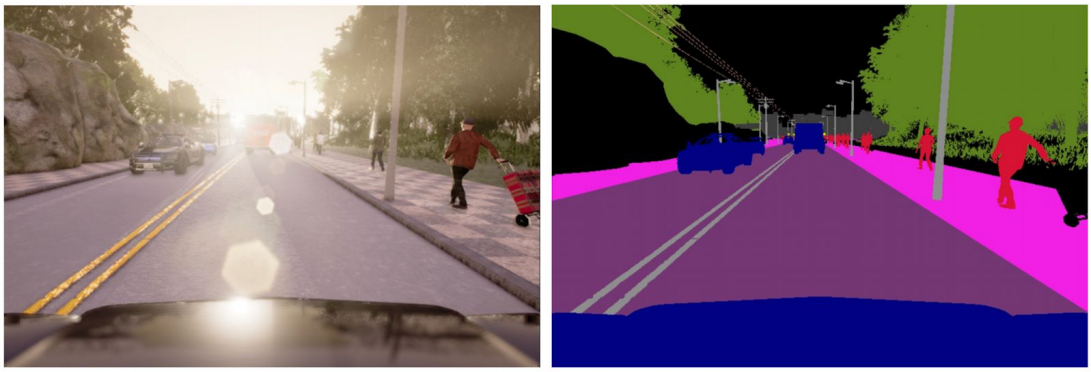

Figure 14: Visualization of sensor view within CARLA [38]

3.3 Lane Navigation and PID control:

The simplest case for a self driving car is to solve the problem of keeping a car in the center of a lane. When the car is too far left, it will steer right, and when it is too right, it will steer left. The intuition involves actuators (steering wheel and pedals), state measurement and the knowledge of the desired state.

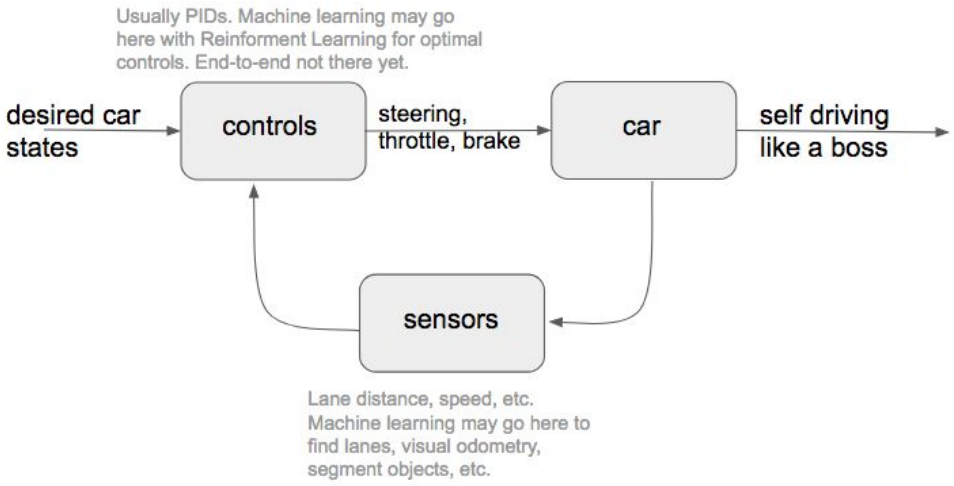

Figure 15: Block diagram of the closed loop controls for self driving cars [37]

Lane Detection: In lane navigation, the first task to complete is the lane detection and locating the position of the car itself. Computer vision is often used through the implementation of classic edge detections to estimate the lanes. Deep learning models are also getting popular, using the data feed from sensors like LiDARs. Lane Detection as important the task is can be bypassed by the lane data feed from GPS, GNSS maps. So, we used the built in HD maps provided with the CARLA simulator to guide our self driving car towards the required direction, and locating its position.

Self-Driving Control: When the car is located and its direction is estimated the task becomes to control it to the desired position and speed. The control of a self driving car can be divided into two parts:

- Longitudinal Control
- Lateral Control

Longitudinal Control: The longitudinal control of the self-driving car dictates all the longitudinal motion of the self-driving car, for example, its longitudinal velocity, acceleration or its longitudinal distance from another preceding vehicle in the same lane on the highway, where the throttle and brakes are the actuators used to implement the control.

Lateral Control: Lateral control of the self-driving car dictates the torque on the steering wheel. The steering system and the lane keeping dynamics are integrated as a whole in the lateral control of the car.

There are currently two approaches to controlling a self driving car.

- AI approach
- Non-AI approach

And a third approach can be a combination of these two.

Here, we focused on the non-AI approach to control our self-driving car. We used a PID based control system to simulate our self-driving over the CARLA simulator.

Proportional Integral Derivative(PID) Controller [36]:

A PID controller dictates the control of a system using three parameters, K_P, K_I and K_D, which are proportional gain, integral gain and derivative gain respectively, often denoted as only P, I and D. Here, the proportional controller deals with the instantaneous errors, whereas the integral controller deals with the overall error till time and the derivative controller with the derivative of the errors. We use the PID controller for both Longitudinal and Lateral control of the car.

3.3.1 Longitudinal Control using PID controller:

The PID controller decides the throttle and break of the self driving car. For the controller to maneuver the motion of the car, it only needs to know the current and target speed of the car, which we can get from the speed sensor attached to the car provided by the CARLA simulator.

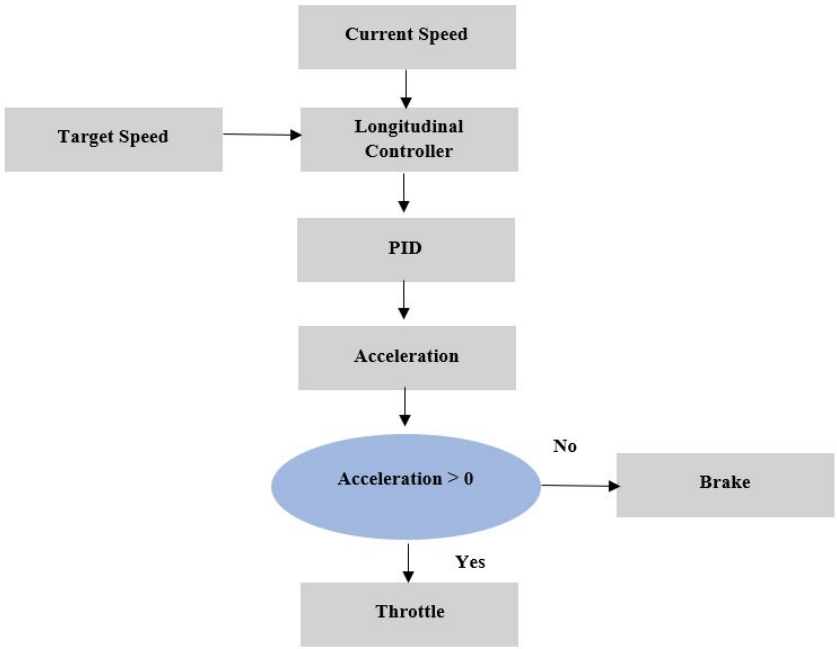

Figure 16: Longitudinal control flowchart

Inside the PID controller, acceleration is calculated as output based on the K_P, K_I and K_D parameters.

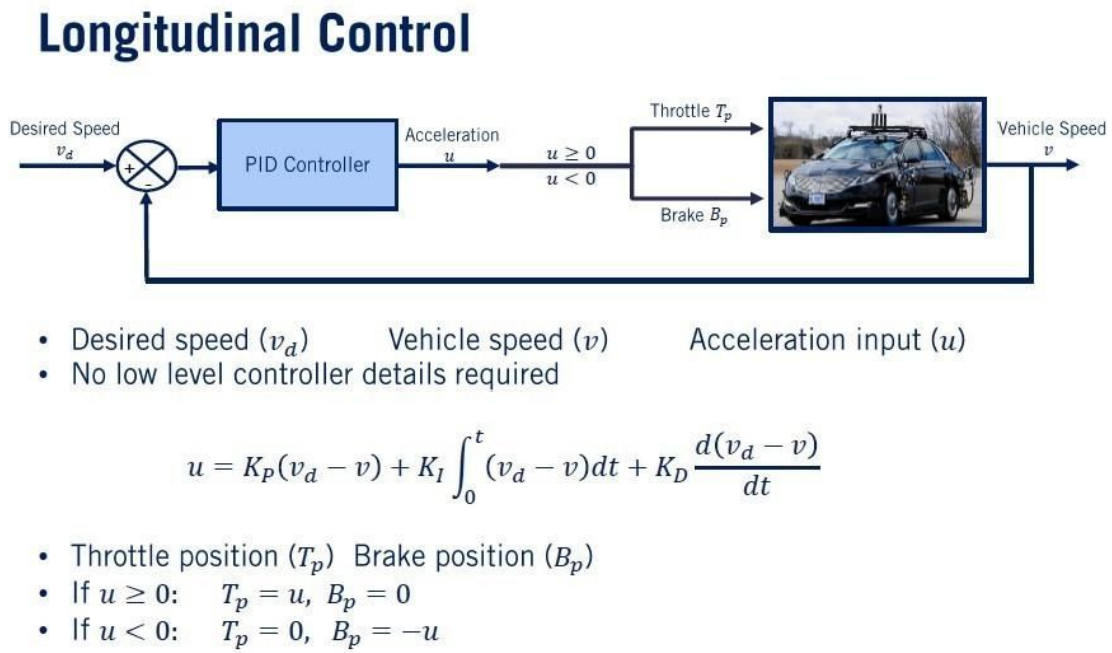

Figure 17: Longitudinal control Image source [11]

3.3.2 Lateral Control using PID controller:

Lateral controller dictates the steering of the self-driving car. For the purpose of lateral control, the controller needs to be provided with the past steering of the car and the target position, which can be termed as the waypoint. CARLA simulator provides us with the waypoints from its built in HD map.

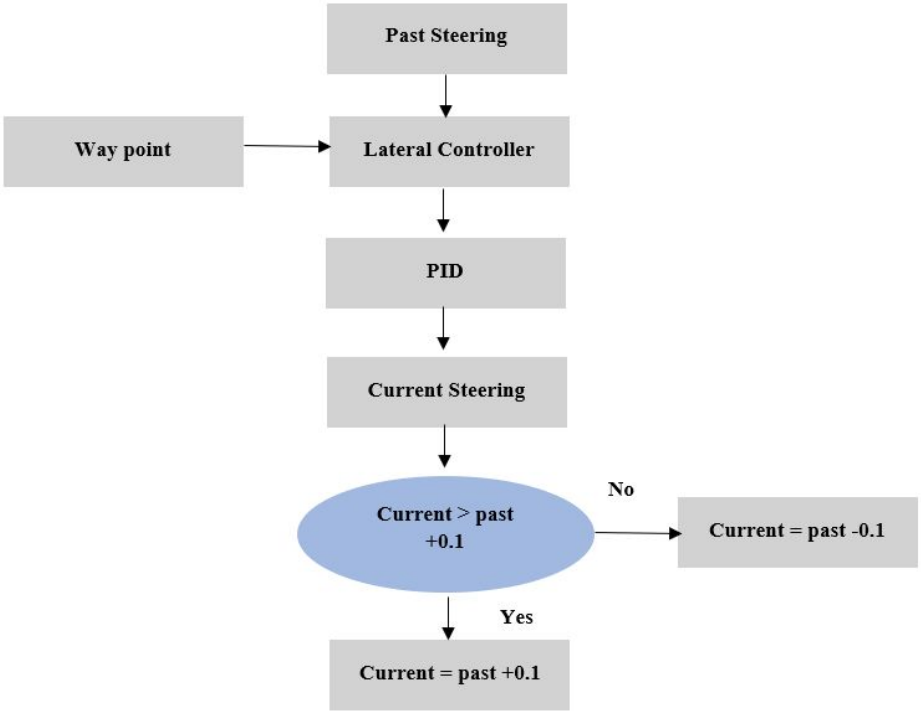

Figure 18: Lateral Control flowchart

PID controller for lateral control works the same way as the longitudinal control, with only exceptions of, there are two errors being monitored here. One is the cross track error, and the other is the heading error.

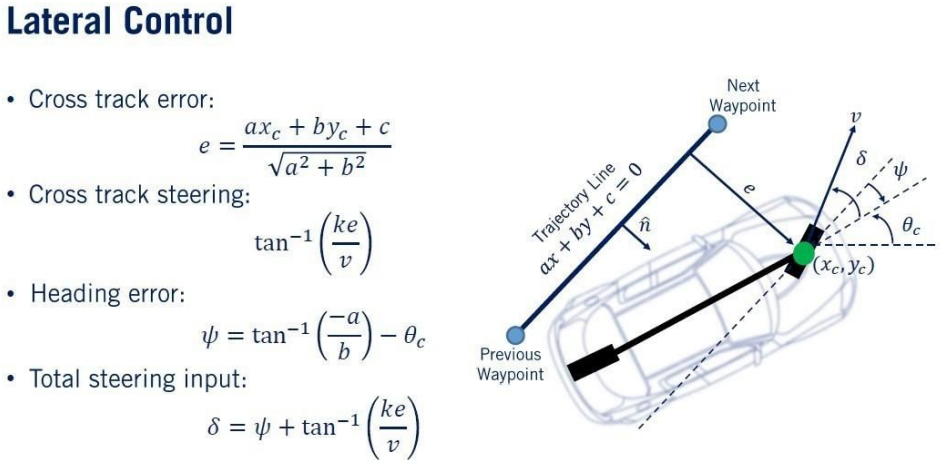

Figure 19: Lateral control (Image source [11])

3.4 Image Resolution Enhancement:

We use SRGAN model here to enhance our image resolution before passing it to the YOLOv5 to get better result. Super Resolution Generative Adversarial Network (SRGAN) [25] is a GAN based image resolution enhancement model. To begin let's first start with GAN.

3.4.1 Generative Adversarial Network(GAN):

GANs generally consist of two networks, Generator and Discriminator [27]. The Generator here tries to create new data, fitting the distribution of training data. The Discriminator on the other hand discriminates between two or more classes of data. And in the GAN, the Generator keeps on generating new data and the Discriminator decides if this generated data is valid or not. The Discriminator here guides the Generator to creating more realistic data in accordance to the training dataset.

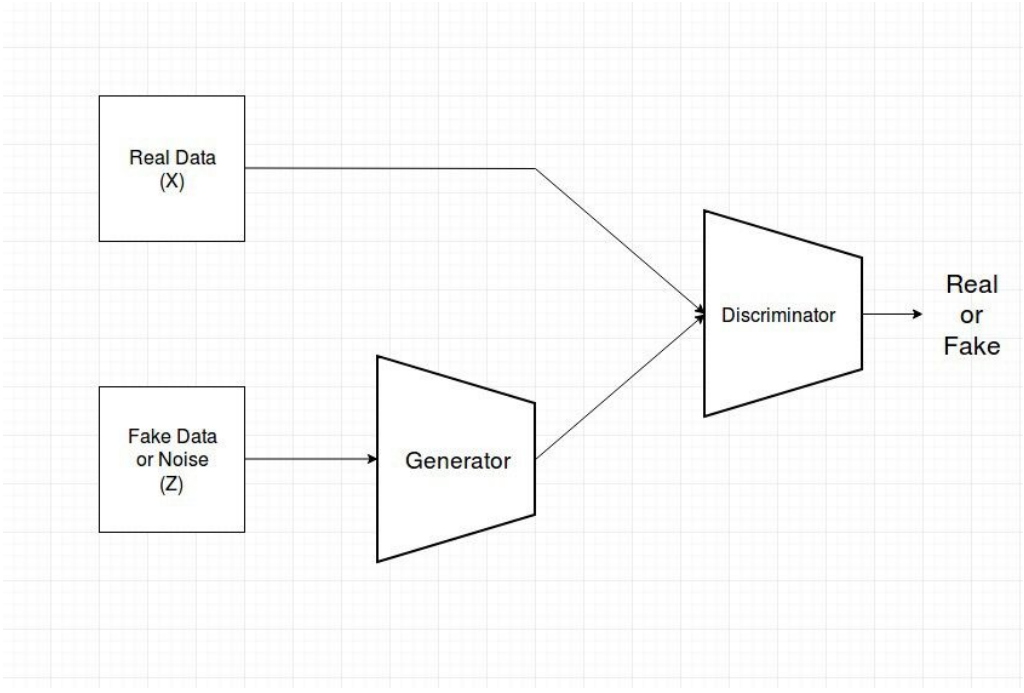

Figure 20: GANs basic architecture [39]

3.4.2 Super-Resolution GAN (SRGAN):

Super-Resolution GAN is a GAN based model, where the model enhances the resolution of an image by manifold. The principal motivation behind SRGAN is to recover finer texture of details from a low resolution image so that the image doesn't seem distorted. SRGAN also consists of two networks, Generator and Discriminator, where the Generator takes a low resolution image and generates a higher resolution image from it, and the Discriminator decides if the higher resolution image is valid or not.

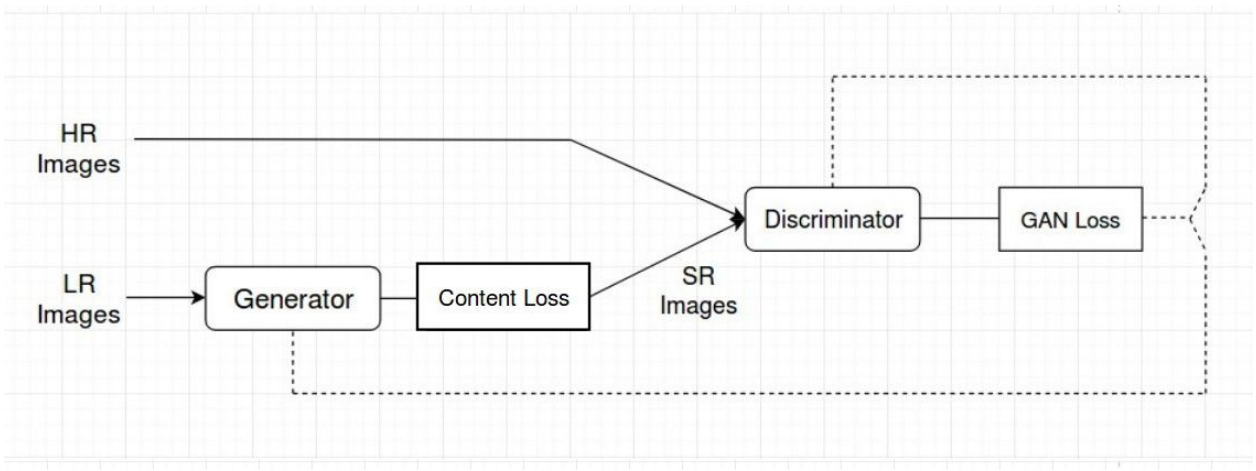

Figure 21: SRGAN architecture [39]

Content Loss: By taking the content loss we are emphasizing perceptual similarity more than the pixel wise similarity which helps us to retain the finer details.

$$l^{SR}_{VGG/i.j} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2$$

Adversarial Loss: This differentiates between the generated super resolution image and the original high resolution image.

$$l_{Gen}^{SR} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR}))$$

The Generator network consists of 16 residual blocks, as the network is much deeper. Both Generator and Discriminator mostly consist of convolution layers, batch normalization. Here Generator uses parameterized RELU whereas the Discriminator uses Leaky RELU mostly.

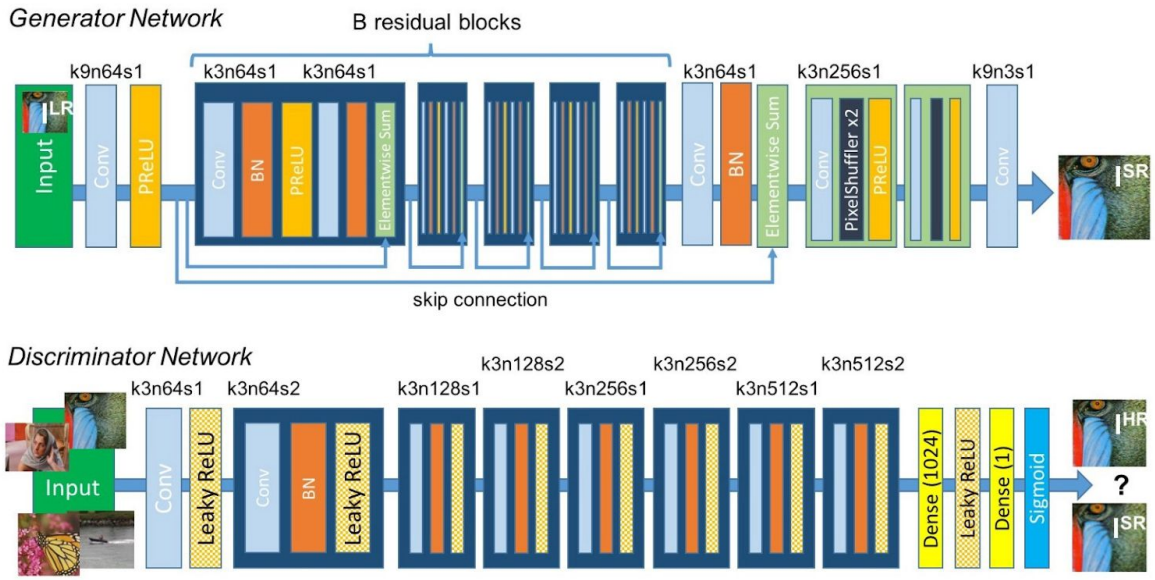

Figure 22: Generator and Discriminator Network Architecture [25]

3.4.3 Implementation of SRGAN:

There are a lot of techniques to enhance an image, interpolation being one of the most common. These techniques are relatively easier to implement but results in distorted images. As for self-driving cars, perception of the environment is a very crucial issue, we used SRGAN, as it gives a very close to real super resolution images, and is also fast for the residual blocks used. For the scarcity of traffic images, we used a more general COCO dataset 2017 [29] to train the model.

Optimizer: Adam optimizer with $\beta 1 = 0.9$, $\beta 2 = 0.999$, $\varepsilon = 1e\text{-}8$ and learning rate 1e-4.

Number of Iterations: 2550

Batch size: 8

3.5 Object Detection using YOLOv5:

YOLOv5 is a smart convolutional neural network (CNN) for real-time identification of objects. The algorithm covers the entire image with a single neural network and splits the image between regions and predicts boundary boxes and possibilities in every region. Fig explains the steps of YOLOv5 algorithm:

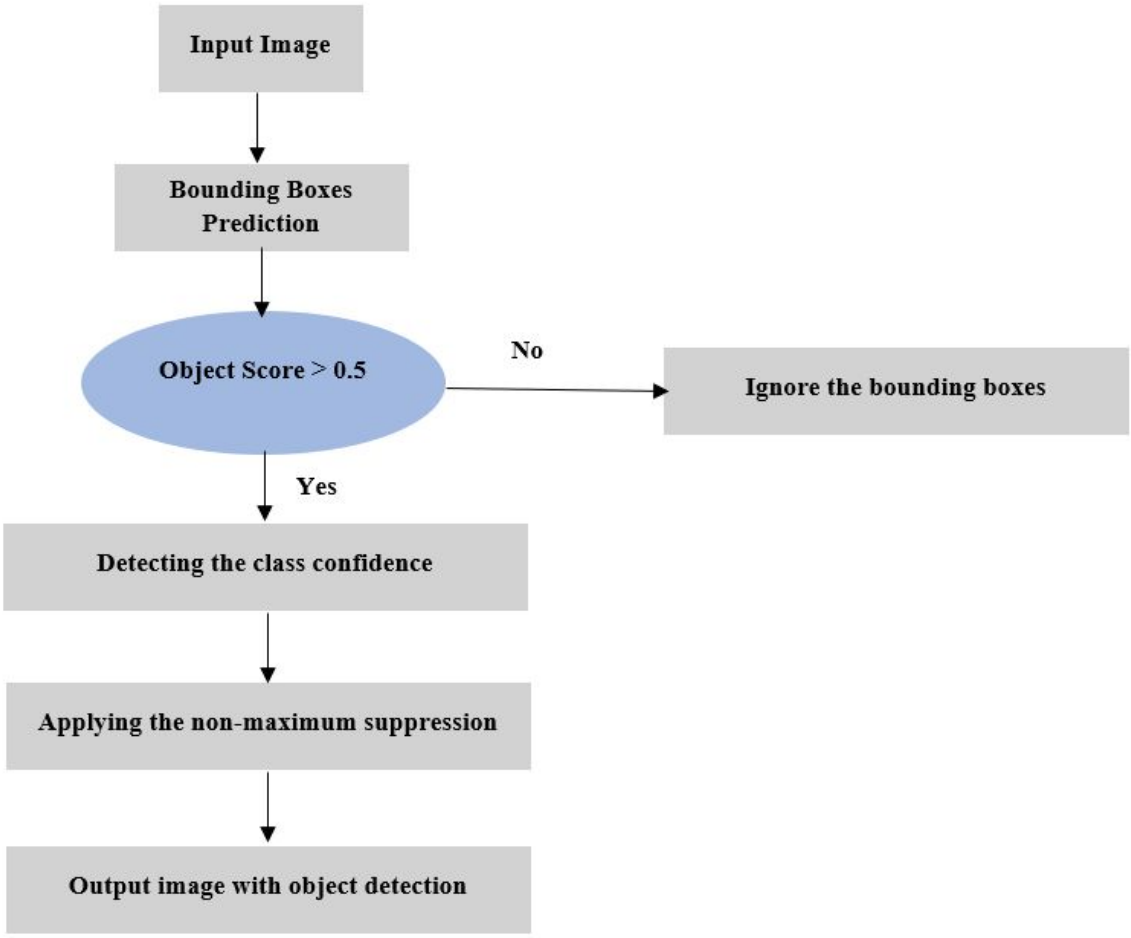

Figure 23: Steps of YOLO algorithm

YOLOv5 comes with 4 architectures - s (small), m (medium), l (large) and xl (extra large). Our test run was based on l and xl, and the final model runs the l architecture. For training, we used pretrained weights based on the coco dataset which contains 80 classes.

For the purpose of preprocessing of the dataset [15], we applied the Super Resolution GAN model and Contrast Limited Adaptive Histogram Equalization (CLAHE) to equalize images. For the purpose of data augmentation, we used traditional techniques, like cropping and flipping of the image. We only flipped the images horizontally, as vertically flipped images will lead to unrealistic dataset.

The modified dataset has been divided into the following numbers:

- No. of training images 7293
- No. of validation images 3433
- No. of test images 450

For the test runs the model was trained up to 80 epochs.

Batch sizes 8 for YOLOv5l and 4 for YOLOv5xl were employed.

Image size - 1024x1024

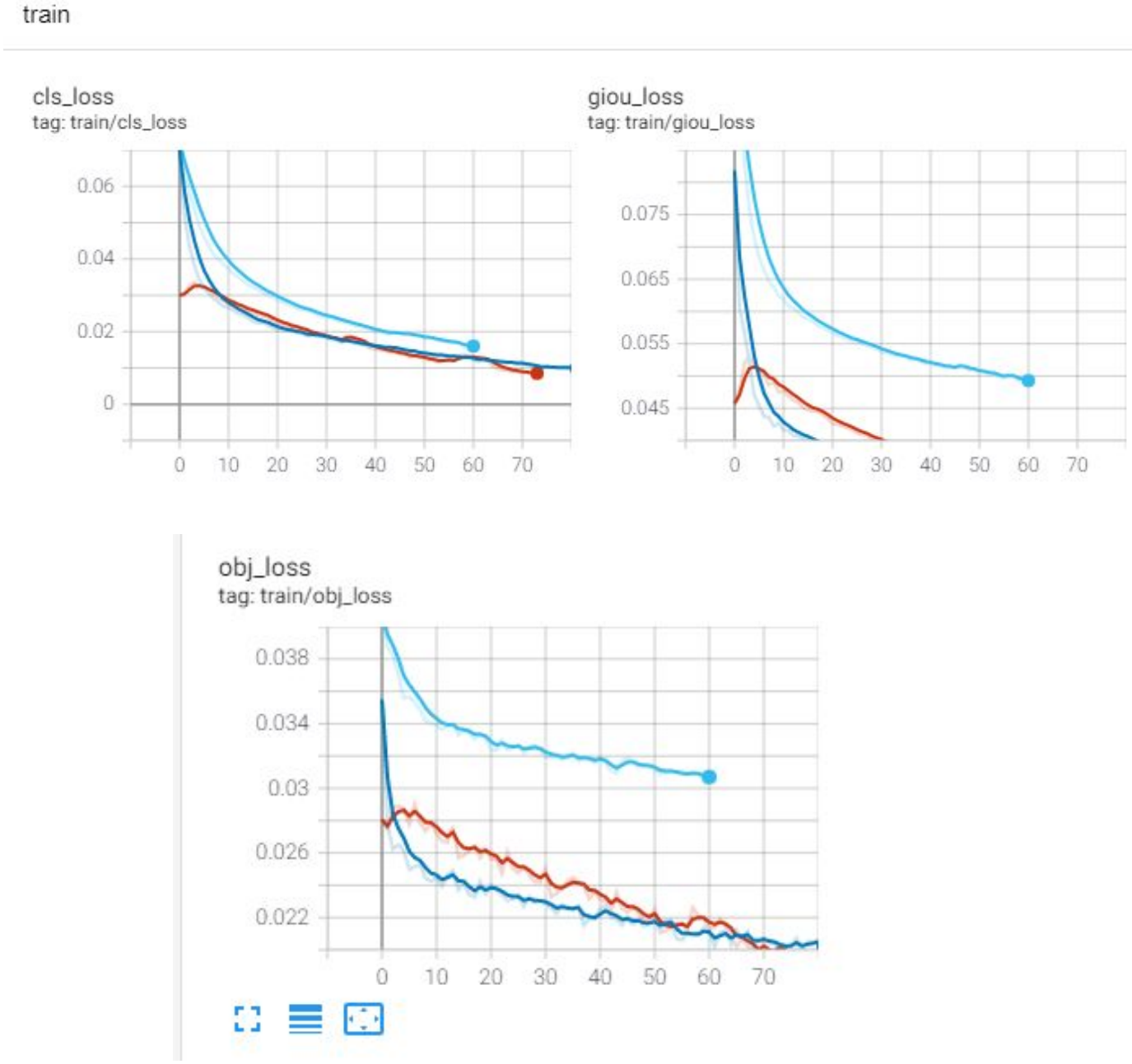

Figure 24: Loss curves for test run training

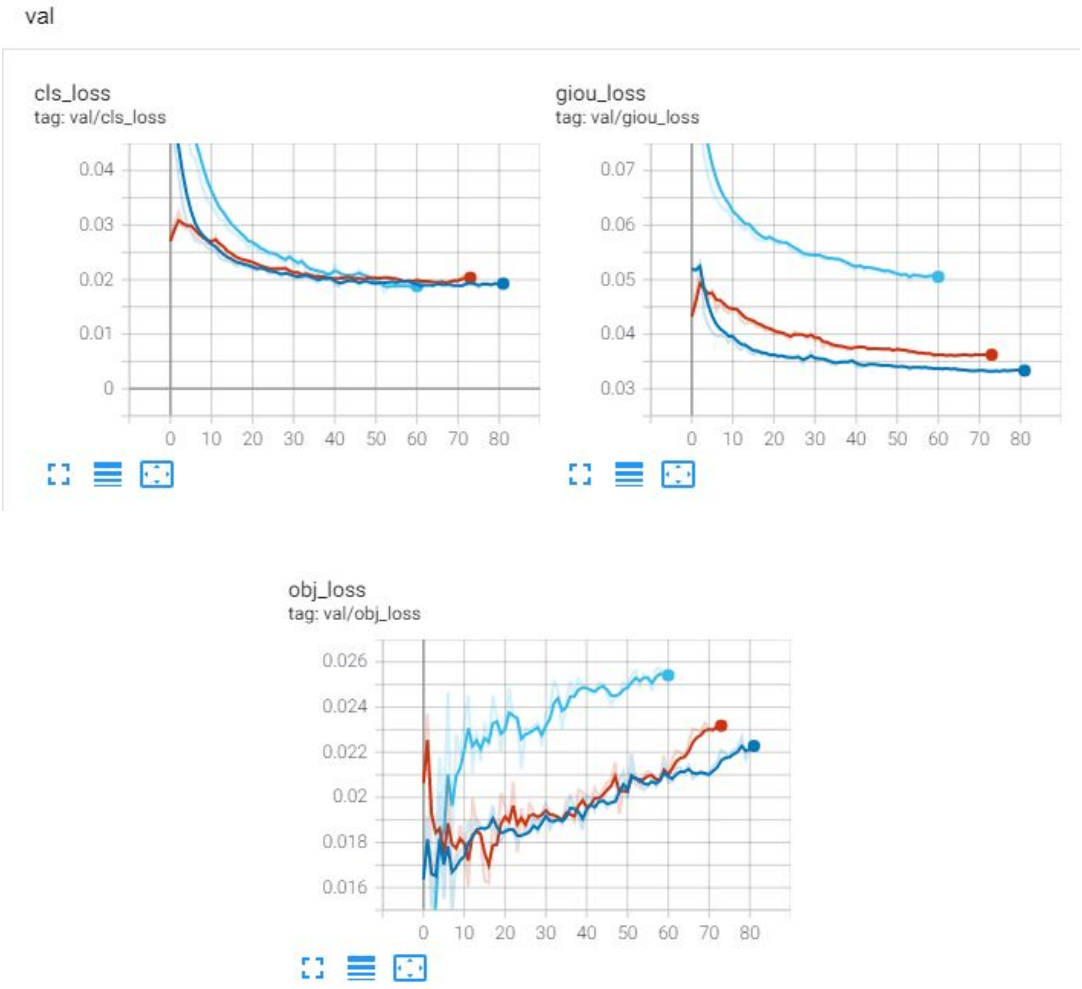

Figure25. Loss curves for test run validation

Figure 26: Histogram of 21 classes generated for the test runs.

Now for the final model, the hyperparameters were similar, except for the epoch number of 267. Though we intended to reach 1000 epochs, it was not possible due to limitation of resources.

Model Summary of the YOLOv5l architecture: 335 layers, 4.7501e+07 parameters, 4.7501e+07 gradients. The architecture consists of a mix of conv, bottleneckCSP, convcat, upsampling, spp layers and a final detect layer.

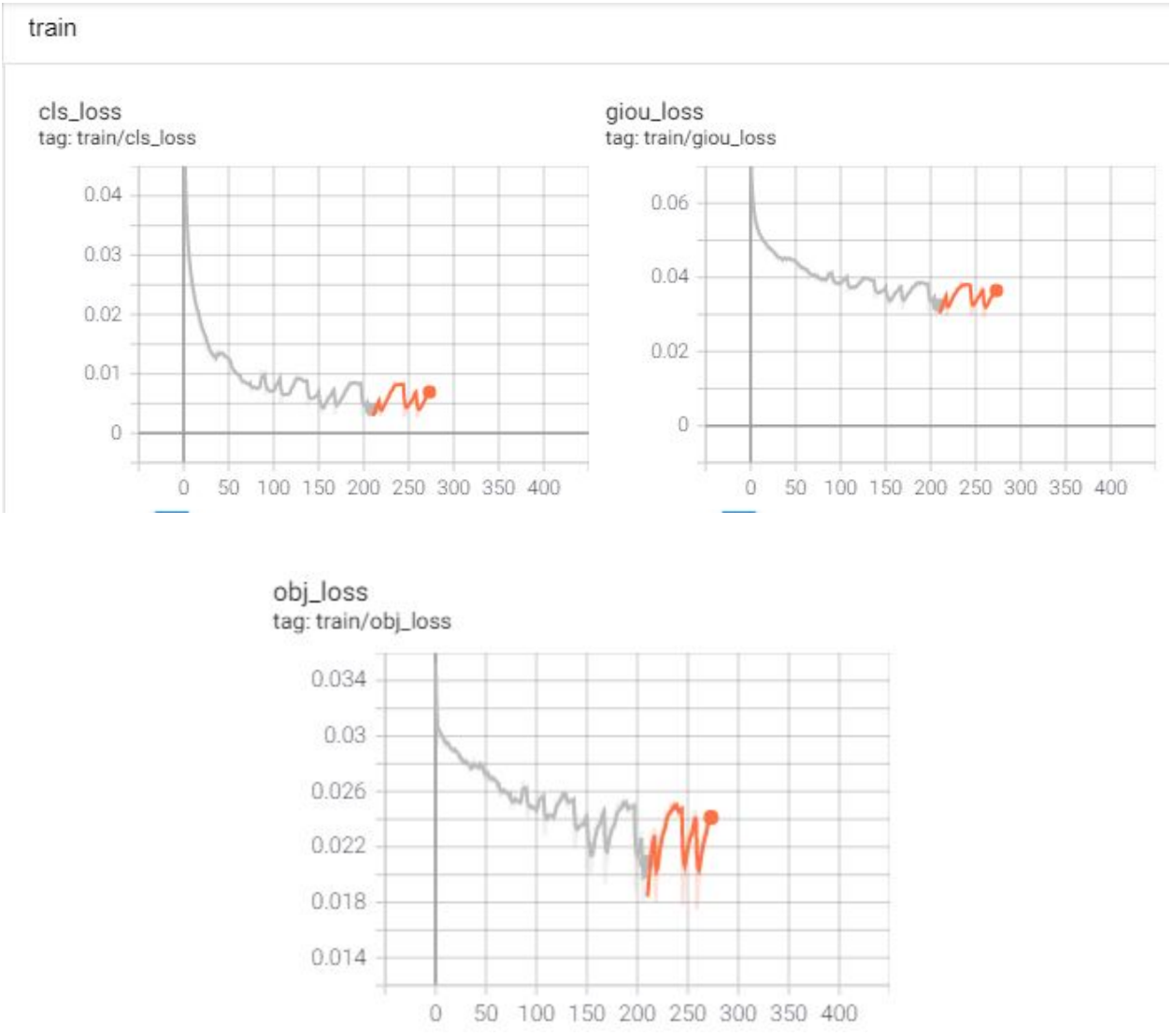

Figure 27: Loss curves for the final model training

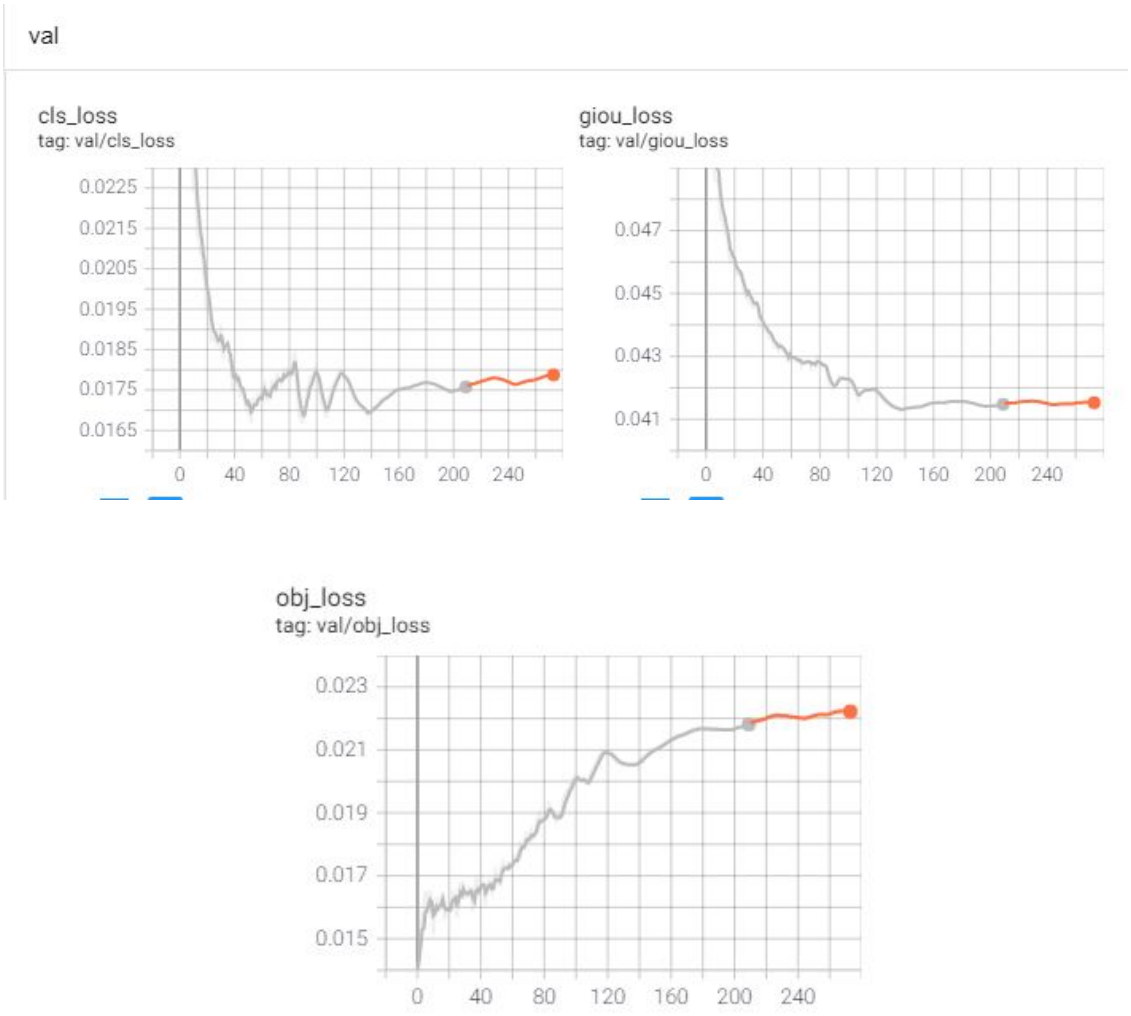

Figure 28: Loss curves for the final model validation

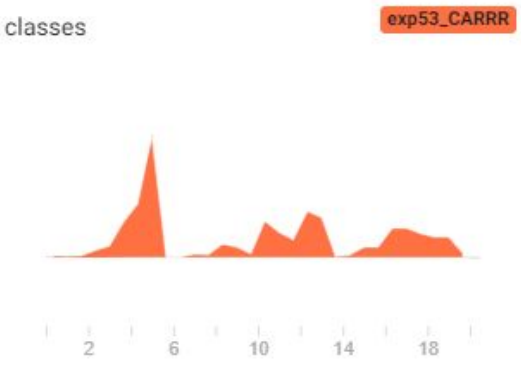

Figure 29: Histogram of detected classes

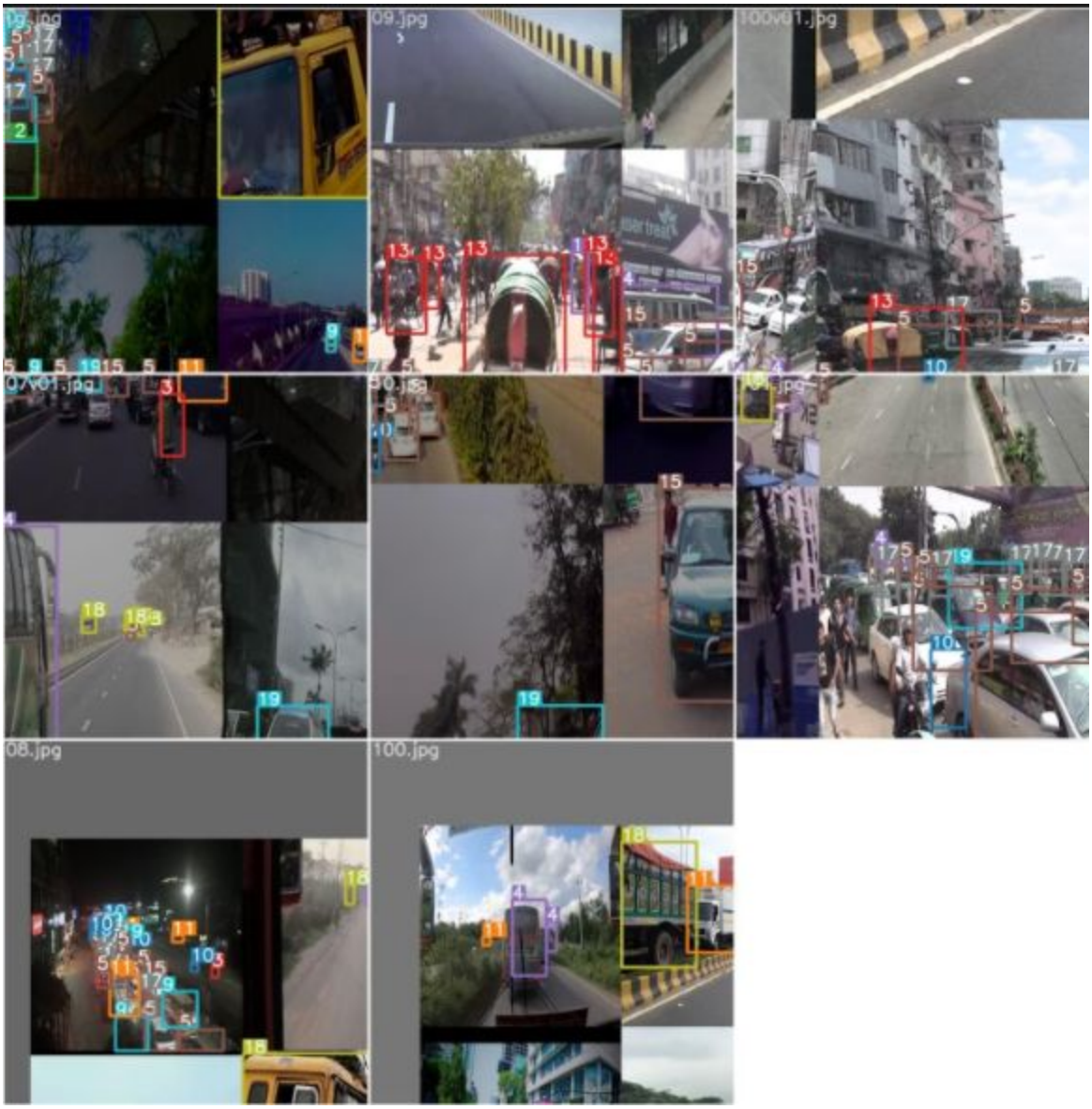

Figure 30: Visualization of the training process

Figure 31: Visualization of the training process with labels

3.6 Use of TensorBoard

TensorBoard is a visualization tool for machine learning applications. It is used for tracking and visualization of performance metrics such as loss and accuracy. Model graphs, histograms, display etc. can also be viewed. Its dashboard is user-friendly and interactive.

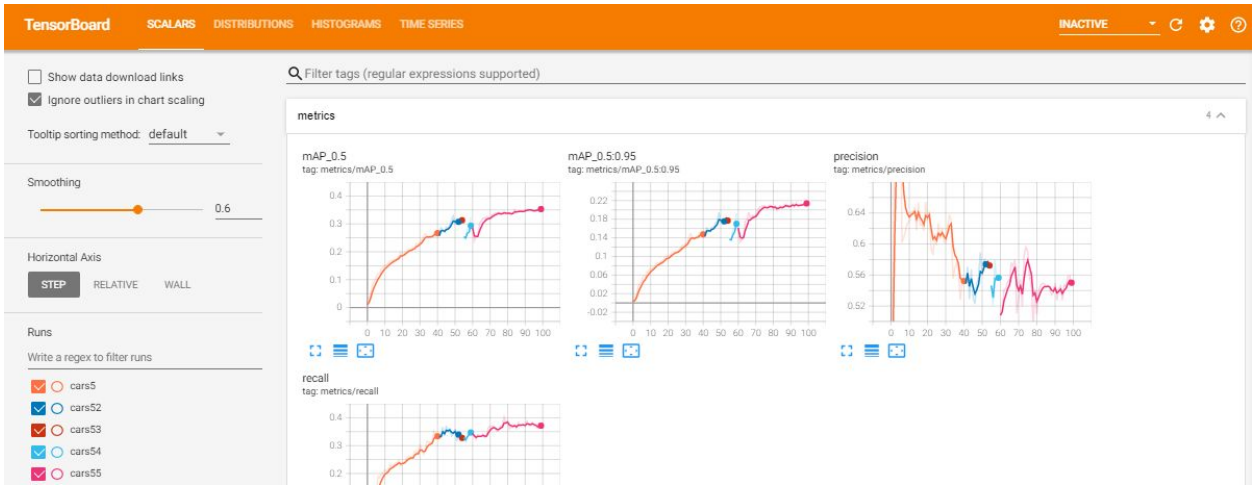

Figure 32: TensorBoard dashboard

The Scalars dashboard shows how the loss and metrics change with every epoch during training. The Graphs dashboard helps visualize the model. The Distributions and Histograms dashboards show a Tensor's distribution over time. This can be useful for visualizing weights and biases and ensuring that they change as intended.

We have used TensorBoard to generate most of the graphs and visualizations in order to gain insights. This process is automated. Simply specifying the directory address is enough to generate desired graphs and visualizations via TensorBoard.

4. Result

Following is the output after applying YOLOv5 for object detection.

Figure 33. Output of the YOLOv5 model

The test runs on YOLOv5 yielded nearly 0.7 mAP@0.5. This is the chosen evaluation metric.

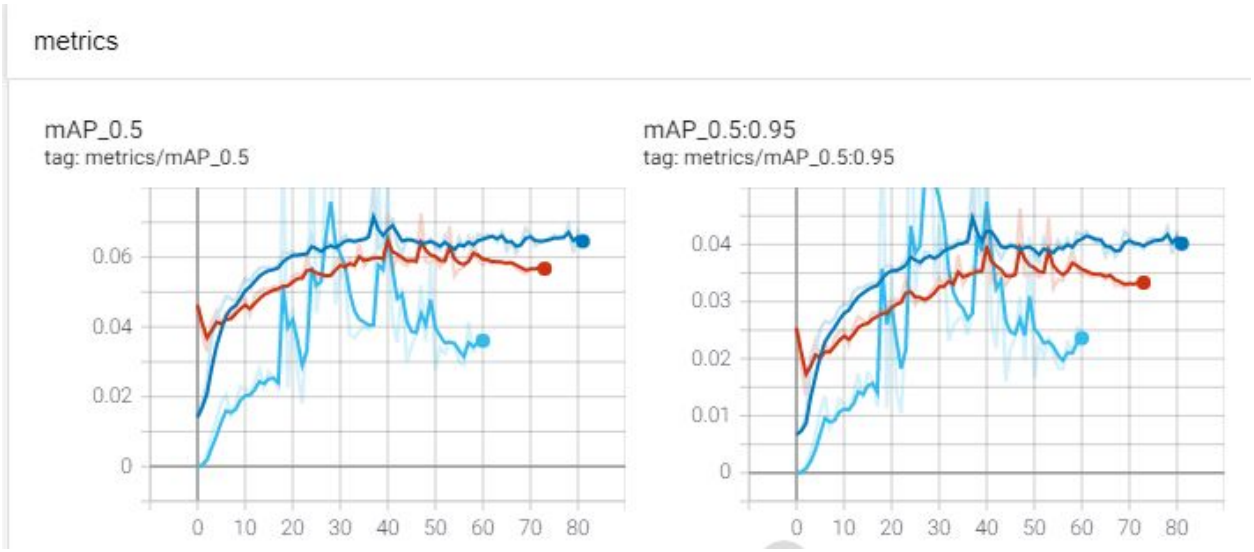

Figure 34: mAP curves for test run validation

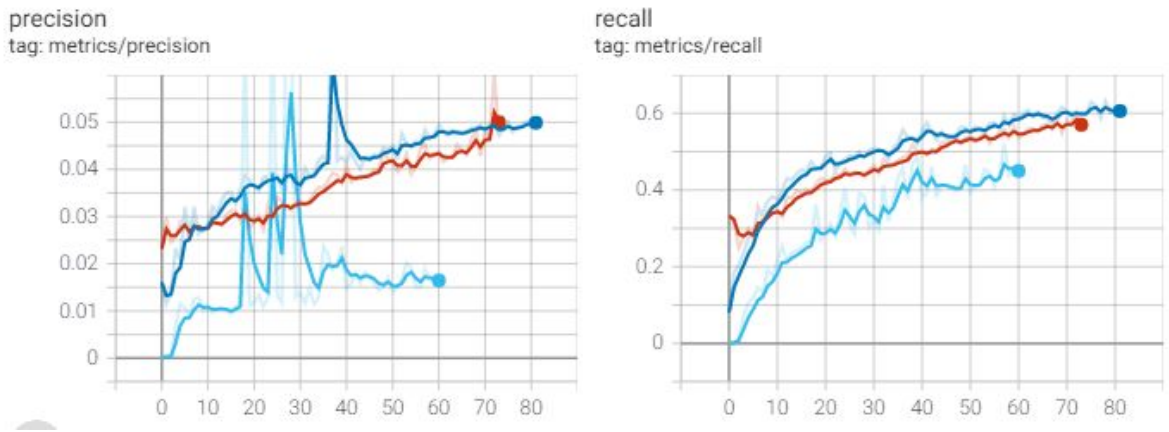

Figure 35. Precision and Recall curves for test run validation

The final YOLOv5l model yielded ~ 0.8 mAP@0.5.

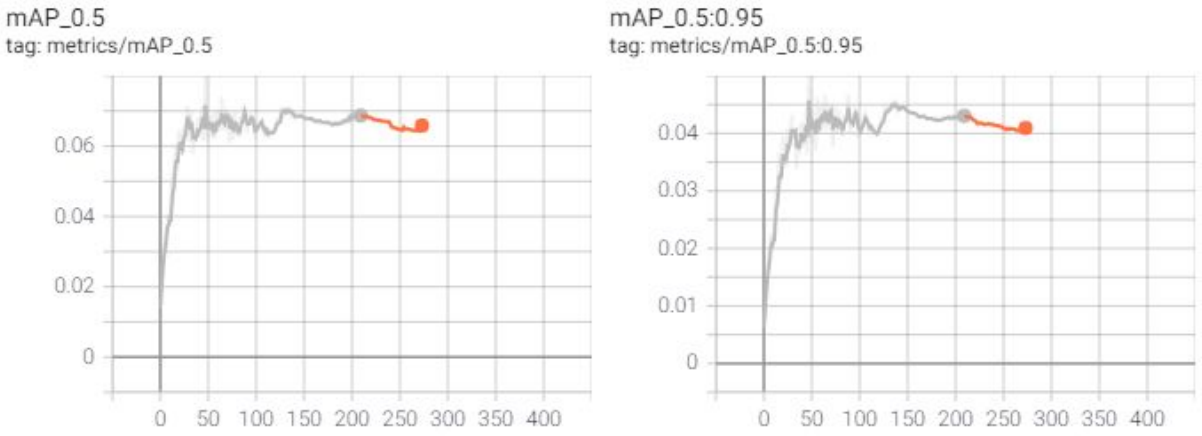

Figure 36: mAP curves for the final model validation

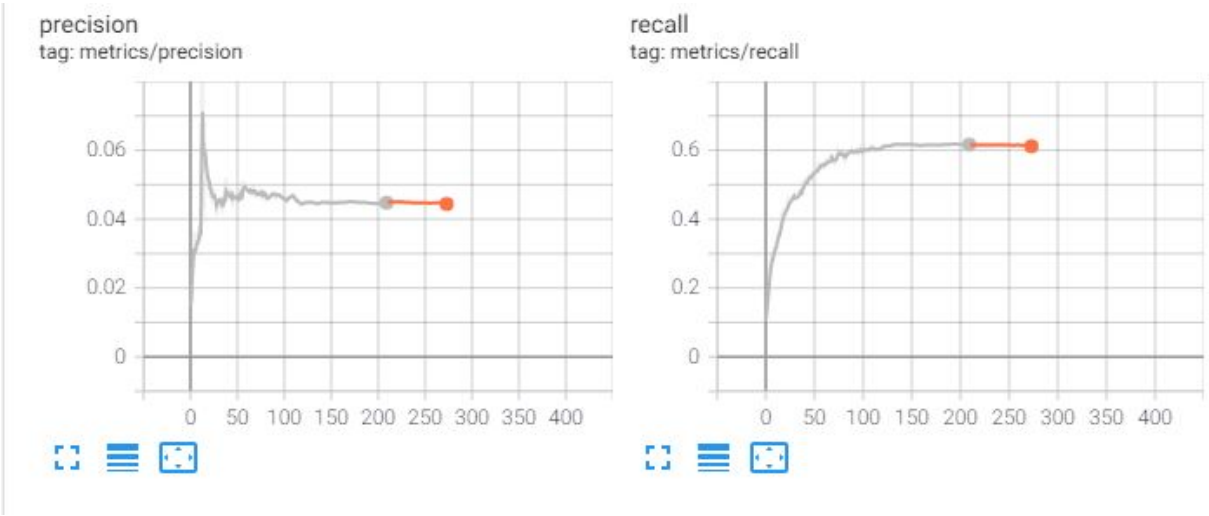

Figure 37: precision and recall curves for the final model validation

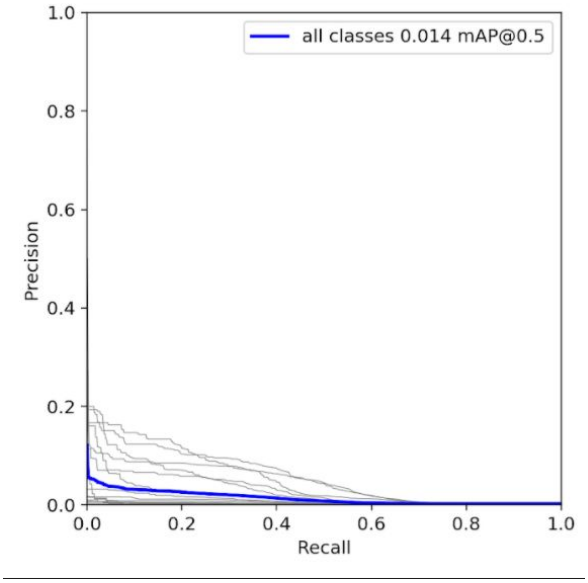

Figure 38: Precision-recall curve of the inference on the test set

From the histogram of predicted labels, we can see that the model can predict some of the elements of even rare classes successfully.

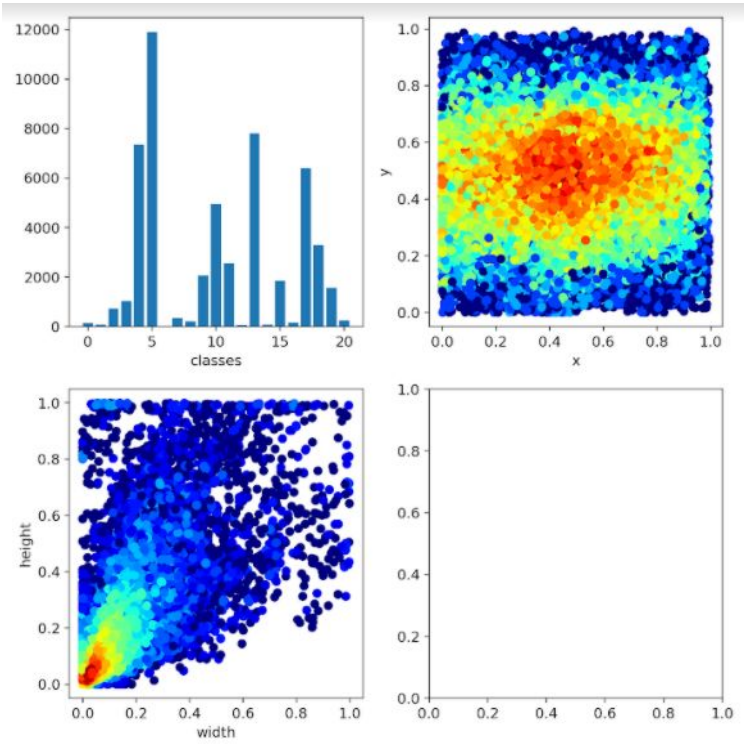

Figure 39. Histogram and scatter plot of labels

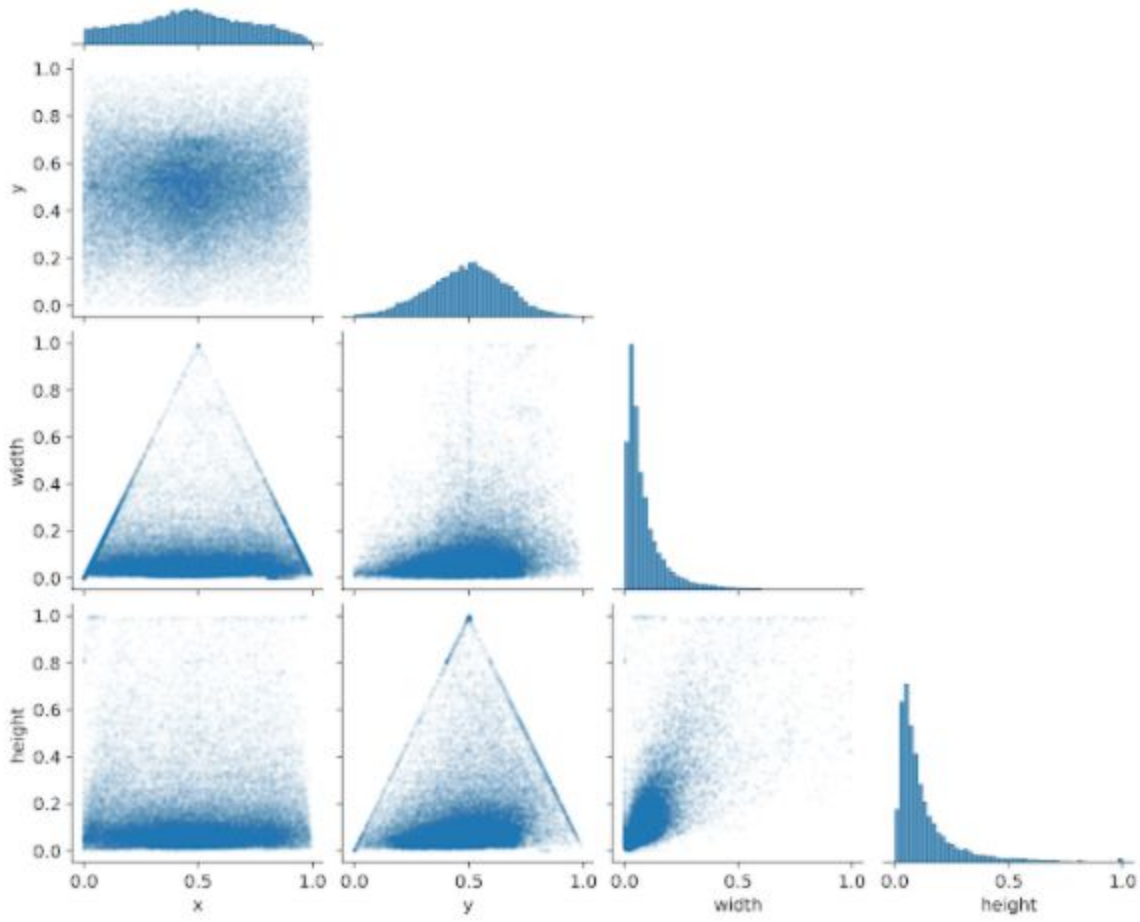

Figure 40. Label correlogram

5. Conclusion

Identification moving objects is crucial and very essential for safeguard and crash avoidance in the case of autonomous vehicles. In our thesis, at the first part between two, in order to guide our own car toward and locate the appropriate directions, we used built-in maps of CARLA simulator. Then we have used PID control for lane navigation. The simulation of detecting and navigating was successful. In the second part of our thesis, we used the YOLOv5 model for object detection. We have achieved an impressive IoU on the Dhaka AI data set. The topic of autonomous vehicles is very new. Among the five stages of autonomy, only three have been achieved worldwide till now. As a result, the resource was difficult to be found during our working period. Nevertheless, with the help of existing resources, we have tried to find out the best possible outcome.

6. Future Development

In our thesis, we did not integrate the two different parts. As mentioned earlier, we have used YOLOv5 for object detection. The outcome of this part can be integrated with the CARLA simulator. For this thesis, we have used built in maps for CARLA simulation. Instead, we can use the YOLOv5 result as the input of the CARLA simulator.

7. References

[1] Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812115

[2] Chenyi Chen, Ari Seff, Alain Kornhauser, Jianxiong Xiao. “DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving”, Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2722-2730 https://openaccess.thecvf.com/content_iccv_2015/html/Chen_DeepDriving_Learning_Affordance_ICCV_2015_paper.html

[3] Manikandasriram Srinivasan Ramanagopal, Cyrus Anderson, Ram Vasudevan, Matthew Johnson-Roberson. “Failing to Learn: Autonomously Identifying Perception Failures for Self-driving Cars”. arXiv: https://arxiv.org/abs/1707.00051v4 [cs.CV]

[4] Christian Häne, Lionel Heng, Gim Hee Lee, Friedrich Fraundorfer, Paul Furgale, Torsten Sattler, Marc Pollefeys. “3D Visual Perception for Self-Driving Cars using a Multi-Camera System: Calibration, Mapping, Localization, and Obstacle Detection”; https://arxiv.org/abs/1708.09839

[5] Jiachen Sun, Yulong Cao, Qi Alfred Chen, Z. Morley Mao; “Towards Robust LiDAR-based Perception in Autonomous Driving: General Black-box Adversarial Sensor Attack and Countermeasures”, https://arxiv.org/abs/2006.16974

[6] Jonah Philion, Amlan Kar, Sanja Fidler; “Learning to Evaluate Perception Models using Planner-Centric MetricsLearning to Evaluate Perception Models using Planner-Centric Metrics”. https://arxiv.org/abs/2004.08745

[7] Der-Hau Lee, Kuan-Lin Chen, Kuan-Han Liou, Chang-Lun Liu, Jinn-Liang Liu; “Deep Learning and Control Algorithms of Direct Perception for Autonomous Driving”; https://arxiv.org/abs/1910.12031

[8] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, Vladlen Koltun' "CARLA: An Open Urban Driving Simulator"; https://arxiv.org/abs/1711.03938

[9] Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, Alexey Dosovitskiy; "End-to-end Driving via Conditional Imitation Learning"; https://arxiv.org/abs/1710.02410

[10] Yudai Nagano, Yohei Kikuta; "SRGAN for Super-Resolving Low-Resolution Food Images." https://dl.acm.org/doi/10.1145/3230519.3230587

[11] Coursera MOOC: "Introduction to Self-Driving Cars" by University of Toronto https://www.coursera.org/learn/intro-self-driving-cars

[12] Larson, William; Zhao, Weihua (2020). "Self-driving cars and the city: Effects on sprawl, energy consumption, and housing affordability". Regional Science and Urban Economics. 81: 103484. doi:10.1016/j.regsciurbeco.2019.103484. ISSN 0166-0462.

[13] How C-V2X in 5G will transform cars and save lives (Analyst Angle) https://www.rcrwireless.com/20200206/analyst-angle/c-v2x-5g-transform-cars-analyst-angle#prettyPhoto

[14] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, ChristopherSTAN, Liu Changyu, … Lijun Yu 于力军. (2021, January 5). ultralytics/yolov5: v4.0 - nn.SiLU() activations, Weights & Biases logging, PyTorch Hub integration (Version v4.0). Zenodo. http://doi.org/10.5281/zenodo.4418161 https://github.com/ultralytics/yolov5

[15] Shihavuddin, ASM; Mohammad Rifat Ahmmad Rashid, 2020, "DhakaAI", https://doi.org/10.7910/DVN/POREXF, Harvard Dataverse, V1

[16] M. Daily, S. Medasani, R. Behringer and M. Trivedi, "Self-Driving Cars," in Computer, vol. 50, no. 12, pp. 18-23, December 2017, doi: 10.1109/MC.2017.4451204.

[17] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, Karol Zieba; "End to End Learning for Self-Driving Cars"; arXiv:1604.07316 [cs.CV]

[18] C. Urmson and W. ". Whittaker, "Self-Driving Cars and the Urban Challenge," in IEEE Intelligent Systems, vol. 23, no. 2, pp. 66-68, March-April 2008, doi: 10.1109/MIS.2008.34.

[19] Gim Hee Lee, Friedrich Faundorfer, Marc Pollefeys; Motion Estimation for Self-Driving Cars with a Generalized Camera; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 2746-2753.

[20] Weihuang Xu, Nasim Souly, Pratik Prabhanjan Brahma; Reliability of GAN Generated Data to Train and Validate Perception Systems for Autonomous Vehicles ; https://openaccess.thecvf.com/content/WACV2021W/AVV/papers/Xu_Reliability_of_GAN_Generated_Data_to_Train_and_Validate_Perception_WACVW_2021_paper.pdf

[21] D. N. -N. Tran, H. -H. Nguyen, L. H. Pham and J. W. Jeon, "Object Detection with Deep Learning on Drive PX2," 2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia), Seoul, Korea (South), 2020, pp. 1-4, doi: 10.1109/ICCE-Asia49877.2020.9276954.

[22] The 3 basic elements to building self-driving cars https://www.cnbc.com/2017/01/05/the-3-basic-elements-to-building-self-driving-cars.html

[23] J. Kocić, N. Jovičić and V. Drndarević, "Sensors and Sensor Fusion in Autonomous Vehicles," 2018 26th Telecommunications Forum (TELFOR), Belgrade, Serbia, 2018, pp. 420-425, doi: 10.1109/TELFOR.2018.8612054.

[24] R. Domínguez, E. Onieva, J. Alonso, J. Villagra and C. González, "LIDAR based perception solution for autonomous vehicles," 2011 11th International Conference on Intelligent Systems Design and Applications, Cordoba, Spain, 2011, pp. 790-795, doi: 10.1109/ISDA.2011.6121753.

[25] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan

Wang, Wenzhe Shi; “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network”; https://arxiv.org/pdf/1609.04802.pdf

[26] Yahui Liu, Xiaoqian Fan, Chen Lv, Jian Wu, Liang Li, Dawei Ding, An innovative information fusion method with adaptive Kalman filter for integrated INS/GPS navigation of autonomous vehicles, Mechanical Systems and Signal Processing, Volume 100, 2018, Pages 605-616, ISSN 0888-3270, https://doi.org/10.1016/j.ymssp.2017.07.051.

[27] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio; “Generative Adversarial Nets”; https://arxiv.org/pdf/1406.2661v1.pdf

[28] A. Kuramoto, M. A. Aldibaja, R. Yanase, J. Kameyama, K. Yoneda and N. Suganuma, "Mono-Camera based 3D Object Tracking Strategy for Autonomous Vehicles," 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 2018, pp. 459-464, doi: 10.1109/IVS.2018.8500482.

[29] The COCO (Common Objects in Context) dataset; https://cocodataset.org/#home

[30] Smith, Graeme E., and Christopher J. Baker. "Echoic flow for radar and sonar." Electronics letters 48.18 (2012): 1160-1161.

[31] Pendleton, Scott D.; Andersen, Hans; Du, Xinxin; Shen, Xiaotong; Meghjani, Malika; Eng, You H.; Rus, Daniela; Ang, Marcelo H. 2017. "Perception, Planning, Control, and Coordination for Autonomous Vehicles" *Machines* 5, no. 1: 6. https://doi.org/10.3390/machines5010006

[32] Schwarting, Wilko, Javier Alonso-Mora, and Daniela Rus. "Planning and decision-making for autonomous vehicles." *Annual Review of Control, Robotics, and Autonomous Systems* (2018).

[33] The 5 Autonomous Driving Levels Explained
https://www.iotforall.com/5-autonomous-driving-levels-explained

[34] FIVE LEVELS TO AUTONOMY https://www.blickfeld.com/blog/5-automation-levels/

[35] Automated Vehicles for Safety
https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety

[36] Farag, Wael. (2019). “Track Maneuvering using PID Control for Self-Driving Cars”. Recent Advances in Electrical & Electronic Engineering (Formerly Recent Patents on Electrical & Electronic Engineering). 12. 10.2174/2352096512666190118161122. ttps://www.researchgate.net/publication/331475267_Track_Maneuvering_using_PID_Control_for_Self-Driving_Cars

[37] How do self driving cars drive? Part 1: Lane keeping assist
https://medium.com/@edersantana/how-do-self-driving-cars-drive-part-1-lane-keeping-assist-581f6ff50349

[38] Semantic Segmentation to Solve The Lyft Perception Challenge
https://medium.com/@kylesf/udacitylyftchallenge-ebffe71a6b22

[39] Single Image Super Resolution Using GANs — Keras
https://medium.com/@birla.deepak26/single-image-super-resolution-using-gans-keras-aca310f33112#id_token=eyJhbGciOiJSUzI1NiIsImtpZCI6ImU4NzMyZGIwNjI4NzUxNTU1NjIxM2I4MGFjYmNmZDA4Y2ZiMzAyYTkiLCJ0eXAiOiJKV1QifQ.eyJpc3MiOiJodHRwczovL2FjY291bnRzLmdvb2dsZS5jb20iLCJuYmYiOjE2MTUxODE2MzQsImF1ZCI6IjIxNjI5NjAzNTgzNC1rMWs2cWUwNjBzMnRwMmEyamFtNGxqZGNtczAwc3R0Zy5hcHBzLmdvb2dsZXVzZXJjb250ZW50LmNvbSIsInN1YiI6IjExMjk3MTI5Nzk4ODk4MDU3MzE2NCIsImVtYWlsIjoiYWhuYWZhc2lmN0BnbWFpbC5jb20iLCJlbWFpbF92ZXJpZmllZCI6dHJ1ZSwiYXpwIjoiMjE2Mjk2MDM1ODM0LWsxazZxZTA2MHMydHAyYTJqYW00bGpkY21zMDBzdHRnLmFwcHMuZ29vZ2xldXNlcmNvbnRlbnQuY29tIiwibmFtZSI6IkFobmFmIEFzaWYiLCJwaWN0dXJlIjoiaHR0cHM6Ly9saDMuZ29vZ2xldXNlcmNvbnRlbnQuY29tL2EtL0FPaDE0R2pmX2Nnck8xZUhoSFhWTFE2NGZmX1I1ZWZlVWxZTjFUWnBqUS1CN2c9czk2LWMiLCJnaXZlbl9uYW1lIjoiQWhuYWYiLCJmYW1pbHlfbmFtZSI6IkFzaWYiLCJpYXQiOjE2MTUxODE5MzQsImV4cCI6MTYxNTE4NTUzNCwianRpIjoiMDI1Yzk4MzM1NGVjZWQ4Yjk5ZWRhMmFhNDJiZmFlM2I5MmFmMDk1OCJ9.J1Kx62kTLrxVcVeg9V9ScOVqXXFSDCRWkyGD44qEPBfB2F3cZK1meipfww9e_sp8OFYDkXy7BbRmTdJfjvW7DONg3VQj6y17fLUWvK7zm0shwPevLjJo4Xy69VATjcVC5RaJrrztCnHlxWqceg_ZR2A

mUoCCCVqR-Mpo40l4xajCsCnAijs6gf2XUnicfF6hbr4n4-y43eaT6M2yavGrsyMpSB7iq69p0VNBku83Da3q7ddaT3qsU4pOUj_hxpwp8-dmyE851fGeKdpHhHx5pPoxyXjf5Qiw00ePlVFkqq4uyE9jDfRUvhq2_vqdk1nzdw9g-K7_G8PxhvNY9Moy_A