Ain Shams University
Faculty of Computer & Information Sciences
Scientific Computing Department

Wassaly

July 2022
Ain Shams University
Faculty of Computer & Information Sciences
Scientific Computing Department

Delivery Self Driving Car

This documentation is submitted as required for the degree of bachelor's in computer and Information Sciences.

By

Nermin Hussein Mohammed

Nada Hany Ahmad

Adham Rabea Mohamed

Mohamed Ali Elshorafy

Muhammad Ali Radwan

Under The Supervision of

[Supervisor 1]

[Doctor Mohamed Marey],

Scientific Computing Department,

Faculty of Computer and Information Sciences,

Ain Shams University.

[Supervisor 2]

[TA Nada Khaled],

Scientific Computing Department,

Faculty of Computer and Information Sciences,

Ain Shams University.

July 2022

Acknowledgements

All praise and thanks to ALLAH, who provided us the ability to complete this work. We hope this work will be accepted.

We are grateful to our parents and our families who are always providing help and support throughout the whole years of study. We hope we can give that back to them.

We also offer our sincerest gratitude to our supervisors, Prof. Dr Mohammed Marey, who has supported us throughout our thesis with their patience, knowledge, and experience.

Finally, I would like to thank my friends and all the people who gave me support and encouragement.

Abstract

On a mission to better everyday life through robotics, we wanted to serve a purpose which is a safe and a better driving and delivery process. As a result, we put the objective of creating a self-driving delivery car to help in decreasing the number of car's accidents and to help in having a better and safer delivery process.

So, we chose to build a self-driving delivery car, where its mission is to autonomously deliver items from a specific area to another without the need of any drivers. This is done by fusing artificial intelligence models to the video streaming of the car, and training some models to detect and recognize all the surroundings around the car to make it able to make good decisions like humans.

A combination of sensors fusion and advanced perception and control methods have been applied in order to achieve the efficient self driving as the car can dodge obstacles, able to move on the road while staying on its lane lines boundaries, detect if there are pedestrians crossing in front of it, furthermore detect and recognize traffic signs and traffic lights.

A delivery mobile application was implemented, so that through it the client can make his orders and the destination will be sent to the car to follow, and then QR-Code will be sent to the client so he can scan it on the car and get his order.

After finishing the thesis, it is proven that using the new technologies in the right place helps improve people's life, and as we can conclude that each person holding the power of knowledge, must use it for the good of humanity.

Table of Contents

List of Figures

List of Abbreviations

DNN: Deep Neural Network CNN: Convolution Neural Network SAE: The Society of Automotive Engineering GPS: Global Positioning System IMU: Inertial Measurement Unit AI: Artificial Intelligence GUI: Graphical User Interface

Chapter One
Introduction

1.1 Problem Definition

Many car accidents happen daily, which are caused by human negligence as out of the 37,133 deaths in car accidents in 2017, 94% of the crashes were due to human error, because it sometimes can be hard for humans to stay focused, specially when they are exhausted as the process of driving needs a lot of attention, focus and energy, and all that requires a great human effort.

About 3000 people die every day in the world due to collision accidents due to many reasons, including excessive speed of some drivers, traffic congestion, etc.

Higher levels of autonomy have the potential to reduce risky and dangerous driver behaviors. The greatest promise may be reducing the devastation of impaired driving, drugged driving, unbelted vehicle occupants, speeding and distraction, all that will lead to a better and a safe environment.

People with disabilities, like the blind, etc, driving is very hard or impossible for them. Highly automated vehicles can help them and make them live the life they want, without any concerns.

Pandemics like covid-19 made it hard and unsafe for delivery drivers and the clients as it includes dealing and interacting with strangers, and that may affect the delivery driver or the clients.

1.2 Objectives

Less driving. More thriving.

The general objective of the project is reducing the accident rate by introducing the latest technologies and combining them on a mission to better everyday life through robotics to serve a purpose which is a safe and a better driving and delivery process. This is achieved by introducing the delivery self-driving car. Its main objective is to autonomously deliver items from a specific area to another.

Successfully moving from point A to point B determined on its GPS to deliver the orders, considering the following set of small objectives, while moving.

The car should be able to move on the road while staying on its lane lines boundaries, even if curves were ahead, without moving randomly on the road. It also should be able to cross an intersection and change its lane when finding a closed lane.

The car should understand the environment around it, so, for example it should be able to detect any pedestrians crossing on its way, and act to ensure that they are safe, furthermore detect if there are any obstacles on its way to dodge it, etc.

The car should detect and recognize traffic signs like speed limits, stop signs, etc, and make the right decision based on what sign it saw, furthermore detect traffic lights and recognize its color whether it's red, green or yellow.

Constructing a mobile application, that through it the client can make his orders and a destination and a path will be sent to the car to follow to reach to the client. A QR-Code will be sent to the client to scan it on the car and collect his order

Building the car's hardware structure which satisfies the needs of the robot in order to fulfill its desired missions.

1.3 Motivations

The internet's evolution continues. Whether it is online shopping, ordering food, buying gifts, grocery runs, shipping official or personal packages the consumer space is increasingly relying on fast and reliable doorstep delivery.

Incoming 5G communication technology will even further enhance the true power of vehicles.

It's easy and safe to deliver orders during crises and pandemics such as covid-19.

Main factors that could motivate online retailers to adopt AV are Cost, Value of fast delivery and Convenience.

help prevent traffic accidents, free up people's time and reduce carbon emissions.

improve car safety and efficiency.

Many companies use these smart systems in their manufacture and improvement, and think about full autonomous driving without human intervention, so many experiments are made and it is expected that by 2025, they will be real in modern cities.

Some countries are enforcing laws for self-driving cars. In 2012, the state of Nevada, in the United States, issued its first self-driving car license.

1.4 Time Plan

As shown in the figure below, the work did start by the beginning of September 2021, we did start by planning and designing the system architecture and identifying all the components that are required for the project to be implemented, after that by the beginning of October we started working parallel on implementing the self driving car simulation on Carla and implementation of The Mobile application.By December we started working on Self driving Car models. By January we bought the essential hardware components ranging from sensors to required electronic components along with the micro-computer. By April, we were working on the distributed system and network, and the networking phase started. By June, the autonomous driving system was finished and so we started testing the entire system together and integrating the whole system modules together, creating the robot with complete functionality, and for the last process, which is the testing part, to ensure that everything is working well and meets the requirements.

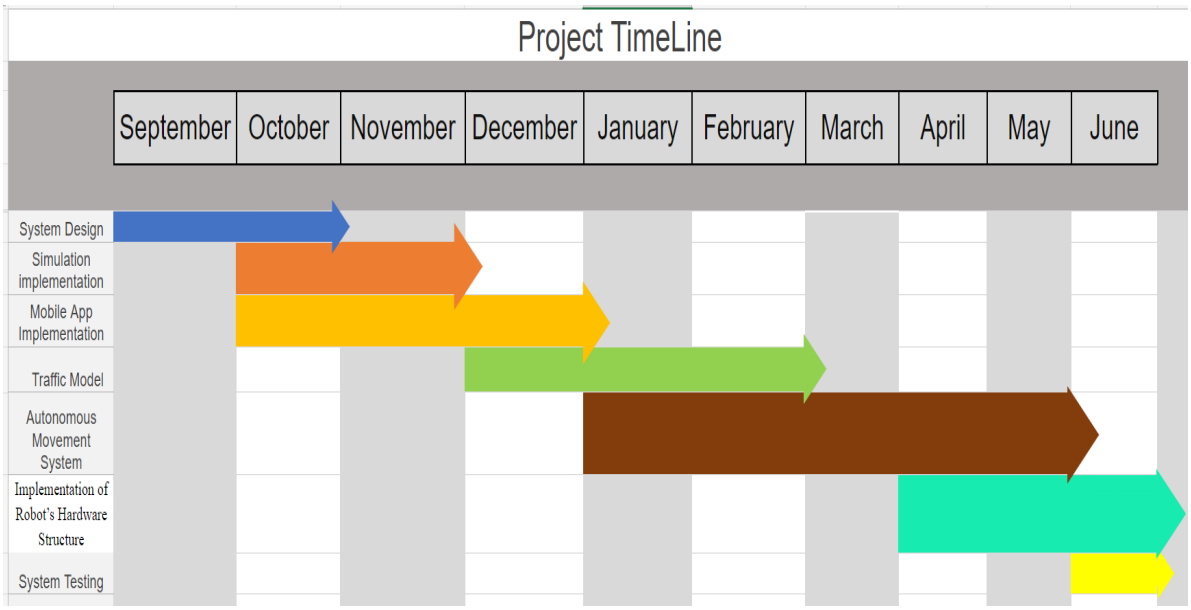


Figure [1-1]

1.5 Documentation Outline

Thesis Statement

The document covers all the project's information and research data from both the literature aspects and the technical aspects.

Chapters Content

Chapter Two will cover the background on which the project is based upon, covering the literature review of the problem introduced, mathematical background, algorithms, and techniques required

Chapter Three will be covering the detailed system architecture and how the system workflow will work, it will also illustrate the coherency between the system modules and components giving a good intuition about the whole process done in the car's system.

Chapter Four will cover the implementation process of the software systems and all the technical details precisely describing each module structure stated in the architecture.

Chapter Five will cover the implementation of the hardware and its components, also the architecture of the network and the distributed system implementation.

Chapter Six will cover the testing steps and how the system can be operated and used.

Chapter Seven will be covering the conclusions from the whole process of creating the car and recommendations for future work that might improve the process in whole.

Chapter Two
Background

2.1 Literature Review

A **self-driving delivery car** is an autonomous robot that provides "last mile" delivery services. An operator may monitor and take control of the robot remotely in certain situations that the robot cannot resolve by itself such as when it is stuck in an obstacle. Self-driving delivery cars can be used in different settings such as food delivery, package delivery, hospital delivery, and room service.

The self-driving delivery car's space has exploded in the past few years, in part due to the global pandemic, the rise of mobile ordering, and the evolution of technology.

The global market for the self-driving delivery cars was valued at $300 million in 2021, according to Quince Market Insights. It's estimated to grow at a compound annual growth rate (CAGR) of 30.3% from now until 2030.

These autonomous delivery vehicles would represent "a significant reallocation of the carrier's costs and would make the service more economical and efficient than with conventional vehicles.

Top brass in the U.S.

Many of the top players are based in the U.S., including Starship, Nuro, and Kiwibot. But there is also plenty of activity overseas. Within one year of launch, Alibaba's XIaomanlv last-mile delivery robots reportedly completed more than 1 million deliveries. And the company has plans to develop 10,000 robots within the next few years.

Legacy players such as Amazon, FedEx, and Postmates are also investing in autonomous vehicles and mobile robots, bringing their systems into more cities and countries regularly. Among the challenges for both sidewalk and street-based systems is a lack of unified regulations across jurisdictions.

Meet the autonomous delivery robots which are shaping the future of food delivery. These could also be used during coronavirus when everybody is staying home.

The Top Delivery Robots :

1. Starship
2. Kiwi
3. Robomart
4. TeleRetail Robot
5. Amazon Scout
6. FedEx SameDay Bot
7. Nuro

This Section will discuss the Amazon Scout Approach and Nuro to autonomous delivery robots.

Nuro is an American robotics company based in Mountain View, California and founded by Jiajun Zhu and Dave Ferguson.[1] Nuro develops autonomous delivery vehicles, and was the first company to receive an autonomous exemption from the National Highway Traffic Safety Administration since its vehicles are designed to carry goods instead of humans

Figure [2-1]

At Amazon, they have been hard at work developing a new, fully-electric delivery system – Amazon Scout – designed to safely get packages to customers using autonomous delivery devices. These devices were created by Amazon, are the size of a small cooler, and roll along sidewalks at a walking pace. These devices will be delivering packages to customers in a neighborhood in Snohomish County, Washington

Customers can shop on the Amazon App and enjoy the same delivery options including fast, FREE Same-Day, One-Day and Two-Day shipping for Prime members.

The devices will autonomously follow their delivery route but will initially be accompanied by an Amazon employee. They developed Amazon Scout, ensuring the devices can safely and efficiently navigate around pets, pedestrians and anything else.

Figure [2-2]

2.2 Autonomous Concepts

2.2.1 Autonomous System

The robot's movement is one of the most essential parts in the project and one of the most complicated, as it depends on a detailed well-built architecture to ensure the stability and the efficiency of the robot's movement through the target environment.

The automation levels are presented in five levels based on the SAE "The society of automotive engineering" [6], the first level implies the ability of the vehicle to maintain the lateral movement or the longitudinal movement, the second level implies the ability of the vehicle to maintain its lateral and longitudinal movement together, the third level implies maintaining the lateral and longitudinal movement along with understanding the environment and interacting with it along with predicting the movements of the surrounding cars and pedestrians with the attention of the pilot, while in level four it implies level three but with dispensing of the pilot, and that is what the autonomous system architecture is based upon as the autonomous patrol robot shouldn't need a human pilot to help it navigating through the environment.[7]

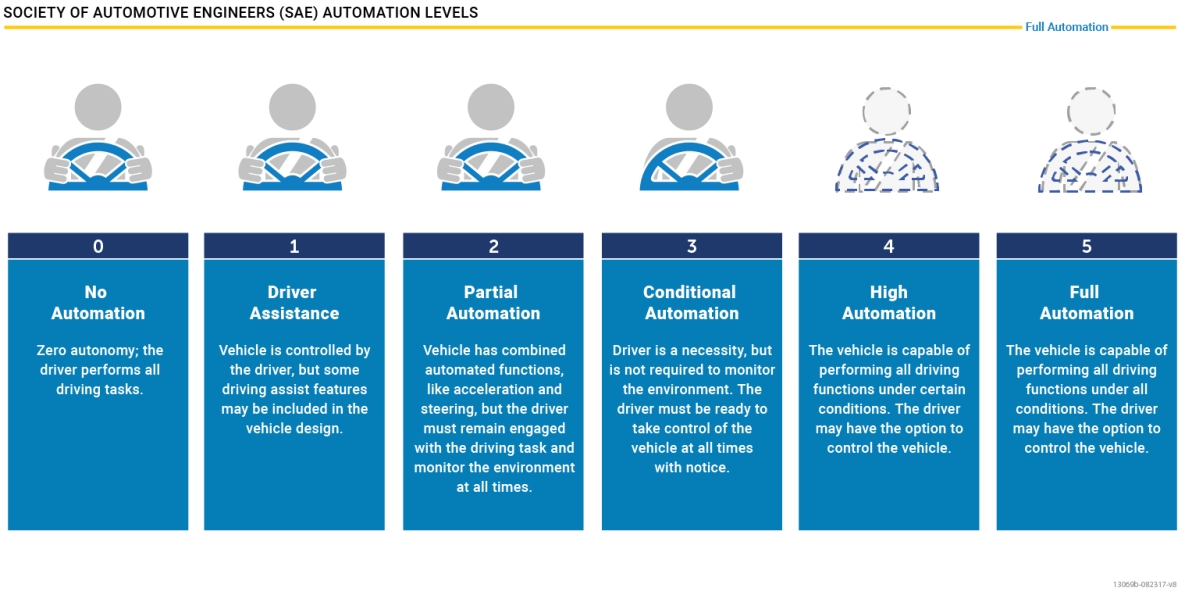


Figure [2-3]

The architecture is divided into main components each contain a specific challenge, the first component is determining the position of the vehicle with

respect to the world, and this challenge is named under the localization problem, the second component is identifying the surrounding environment, detecting the moving objects and estimating their movement, this component is known as the visual perception, the third component is determining the destination and how to navigate through the map using the data from the localization part and the visual perception part, and this is known as the motion planner, lastly the component which translates the motion planner instructions and turn them into orders for the hardware steering and throttling systems in the car to move is the controller.

2.2.2 Localization Problem

The localization module as mentioned is used for estimating the precise location and orientation state of the Robot's location, it uses the sensors inputs for these operations, but it faces big challenges, as for estimating the position, depending on the GPS alone will not be efficient enough as the responses from the GPS takes time between each response, so the system must have real-time estimation of the vehicle's position, so the sensor fusion step takes place, the sensor fusion step is combining between many sensors to get a more accurate result of the target measurement.

For the position estimation sensor fusion, we did use a Kalman filter [8].

In statistics and control theory, Kalman filtering, also known as linear quadratic estimation (LQE), is an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone, by estimating a joint probability distribution over the variables for each timeframe. The filter is named after Rudolf E. Kálmán, who was one of the primary developers of its theory.

The algorithm works in a two-step process. In the prediction step, the Kalman filter produces estimates of the current state variables, along with their uncertainties. Once the outcome of the next measurement (necessarily corrupted with some amount of error, including random noise) is observed, these estimates are updated using a weighted average, with more weight being given to estimates with higher certainty. The algorithm is recursive. It can run in real time, using only the present input measurements and the previously calculated state and its uncertainty matrix; no additional past information is required.

Optimality of the Kalman filter assumes that the errors are Gaussian. In the words of Rudolf E. Kálmán: "In summary, the following assumptions are made about

random processes: Physical random phenomena may be thought of as due to primary random sources exciting dynamic systems. The primary sources are assumed to be independent gaussian random processes with zero mean; the dynamic systems will be linear.

Though regardless of Gaussianity, if the process and measurement covariances are known, the Kalman filter is the best possible linear estimator in the minimum mean-square-error sense.

As our problem is nonlinear, we need an improved version of Kalman filter to perform the sensor fusion allowing us to get the best estimate of the robot's location and rotation states, and here we will use the extended Kalman filter or the EKF. [9]

In estimation theory, the extended Kalman filter (EKF) is the nonlinear version of the Kalman filter which linearizes about an estimate of the current mean and covariance. In the case of well-defined transition models, the EKF has been considered the de facto standard in the theory of nonlinear state estimation, navigation systems and GPS.

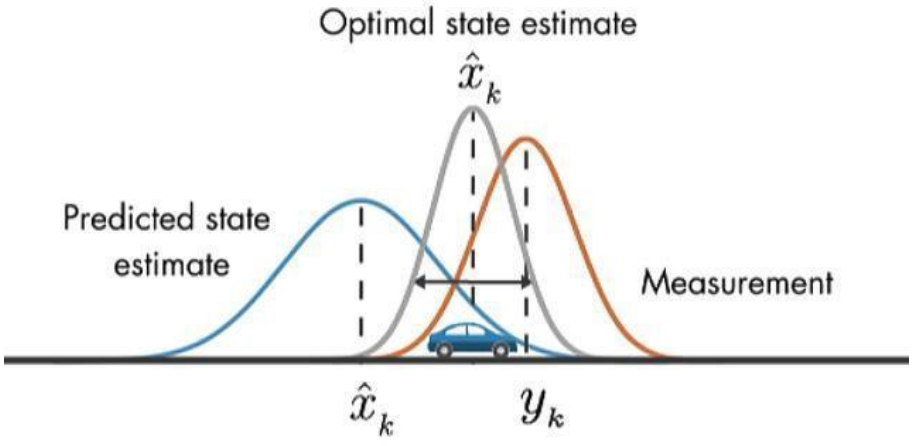


Figure [2-4]

2.2.3 Visual Perception

The visual perception main idea is detecting and recognizing the objects around the world, enabling it to understand more about the surrounding environment. We would use the information around us to pass it to the motion planner, so that we give it the whole scene and all the data required for the component to take the best decision, so for the object detection, we used an algorithm which is SSD “Single Shot Multibox Detection”, it is used to detect each object and localize it. [10]

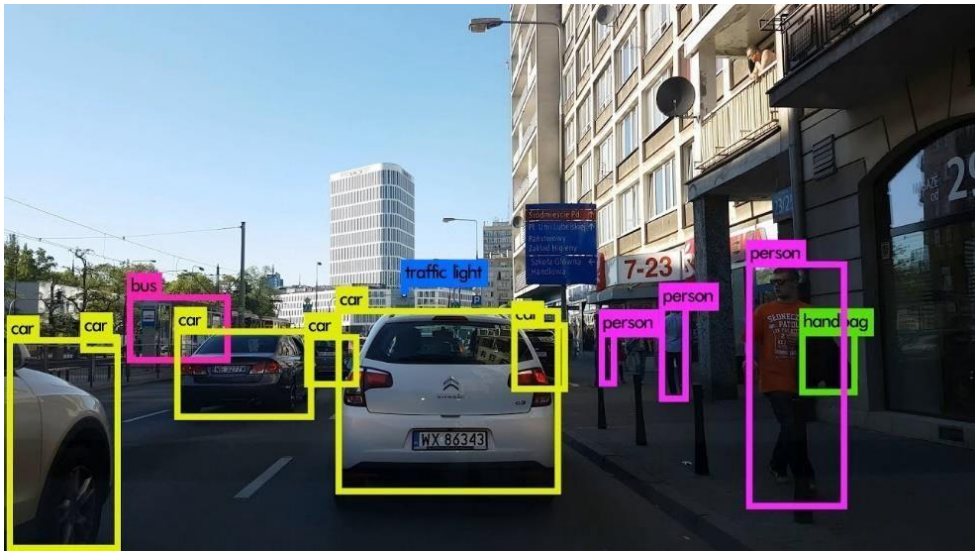


Figure [2-5]

SSD is a method for detecting objects in images using a single deep neural network. Its approach discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. At prediction time, the network generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape. Additionally, the network combines predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes.

The SSD model is simple relative to methods that require object proposals because it eliminates proposal generation and subsequent pixel or feature resampling stage and encapsulates all computation in a single network. This makes SSD easy to train and straightforward to integrate into systems that require a detection component.

Experimental results on the PASCAL VOC, MS COCO, and ILSVRC datasets confirm that SSD has comparable accuracy to methods that utilize an additional object proposal step and is much faster, while providing a unified framework for both training and inference.

Compared to other single stage methods, SSD has much better accuracy, even with a smaller input image size. For 300×300 input, SSD achieves 72.1% mAP on VOC2007 test at 58 FPS on a Nvidia Titan X and for 500×500 input, SSD achieves 75.1% mAP, outperforming a comparable state of the art Faster R-CNN model. [11]

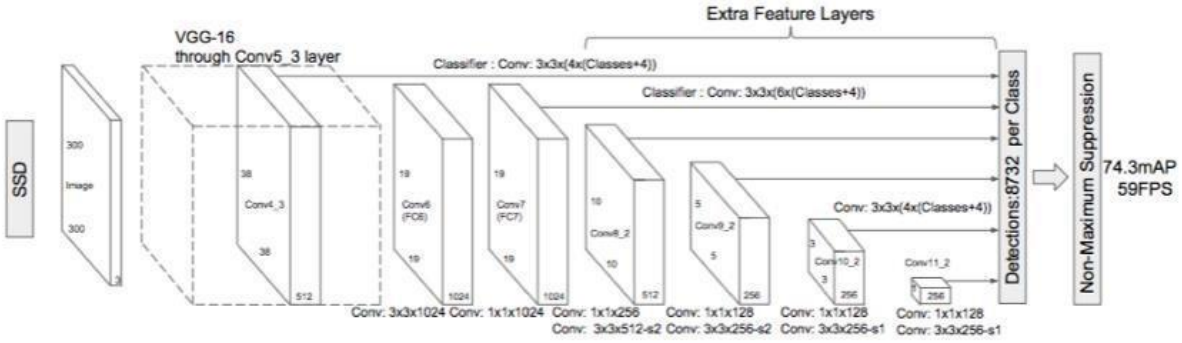


Figure [2-6]

In addition to object detection, semantic segmentation is also essential for the visual perception process, Semantic segmentation is a method of scene understanding in which classification is performed on every single pixel of an image. Semantic segmentation is used in autonomous vehicles to locate frontal objects such as roads, dividers, vehicles, pavements, etc. It is a vital subsystem of the vehicle's navigation system.

Semantic segmentation is a method of scene understanding in which classification is performed on every single pixel of an image. Semantic segmentation is used in

autonomous vehicles to locate frontal objects such as roads, dividers, vehicles, pavements, etc. It is a vital subsystem of the vehicle's navigation system. [12]

Figure [2-7]

Mean shift is a procedure for locating the maxima—the modes—of a density function given discrete data sampled from that function. This is an iterative method, and we start with an initial estimate .

Let a kernel function be given. This function determines the weight of nearby points for re-estimation of the mean. Typically, a Gaussian kernel on the distance to the current estimate is used. The weighted mean of the density in the window is determined.

Although the mean shift algorithm has been widely used in many applications, a rigid proof for the convergence of the algorithm using a general kernel in a high dimensional space is still not known. Aliyari Ghassabeh showed the convergence

of the mean shift algorithm in one-dimension with a differentiable, convex, and strictly decreasing profile function. [13]

However, the one-dimensional case has limited real world applications. Also, the convergence of the algorithm in higher dimensions with a finite number of the (or isolated) stationary points has been proved. However, sufficient conditions for a general kernel function to have finite (or isolated) stationary points have not been provided.

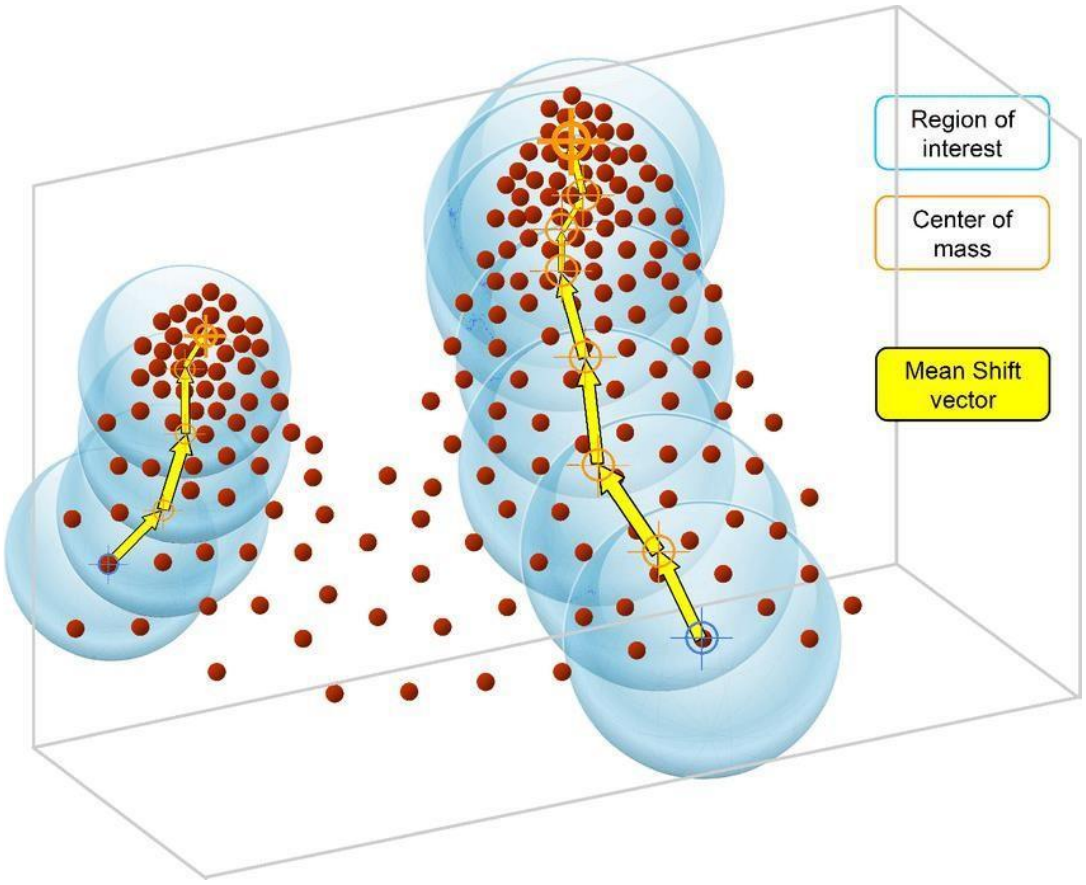


Figure [2-8]

2.2.4 Motion Planning Theory

In our case, this problem is more complicated as there are many parameters to count during the navigation process, including the vehicle's current localization, the environment surrounding the vehicle which is obtained from the visual perception.

Before beginning to use the data from the robot's environment, we should start by getting the initial path that the robot needs to follow, a path is a set of waypoints in the world coordinates, these points will be initially be taken from a mapping provider, the mapping provider uses satellite data to construct a map that contains the roads on the streets for vehicles to be guided with, and then the user must select the starting point and the end point, after that the map provider will generate a set of way points based on the shortest path, and there are several algorithms that are capable of generating such points as A-Star algorithm. [14]

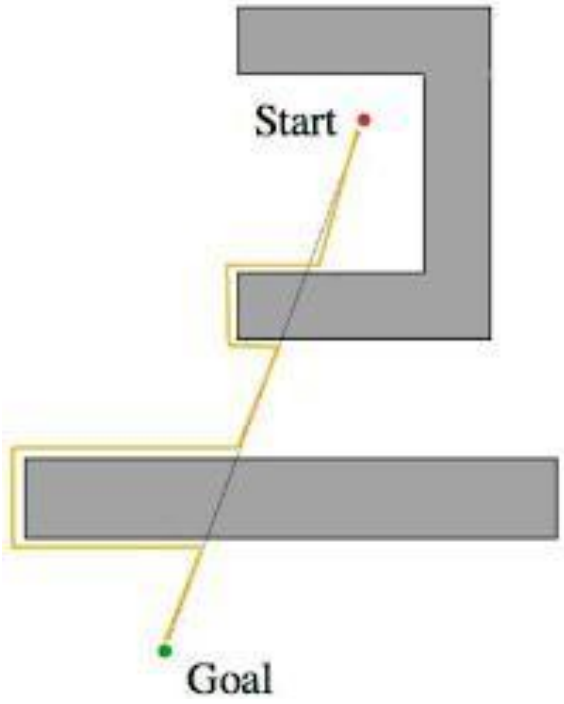


Figure [2-9]

2.2.5 PID Controller

After having completed the previous models and calculated exactly the position velocity, and the rotations of the robot using the localization part and using the visual perception to get a good understanding of the environment combining it with the motion planner to get the target waypoints that the robot wants to move to, we must translate the desired movement that we need to achieve to a meaningful signal to the hardware to start the movement.

Where the PID controller comes, a PID controller is a proportional–integral–derivative controller (PID controller or three-term controller) is a control loop mechanism employing feedback that is widely used in industrial control systems and a variety of other applications requiring continuously modulated control. A PID controller continuously calculates an error value as the difference between a desired setpoint (SP) and a measured process variable (PV) and applies a correction based on proportional, integral, and derivative terms (denoted P, I, and D respectively), hence the name.

In practical terms it automatically applies an accurate and responsive correction to a control function. An everyday example is the cruise control on a car, where ascending a hill would lower speed if only constant engine power were applied. The controller's PID algorithm restores the measured speed to the desired speed with minimal delay and overshoot by increasing the power output of the engine. [15]

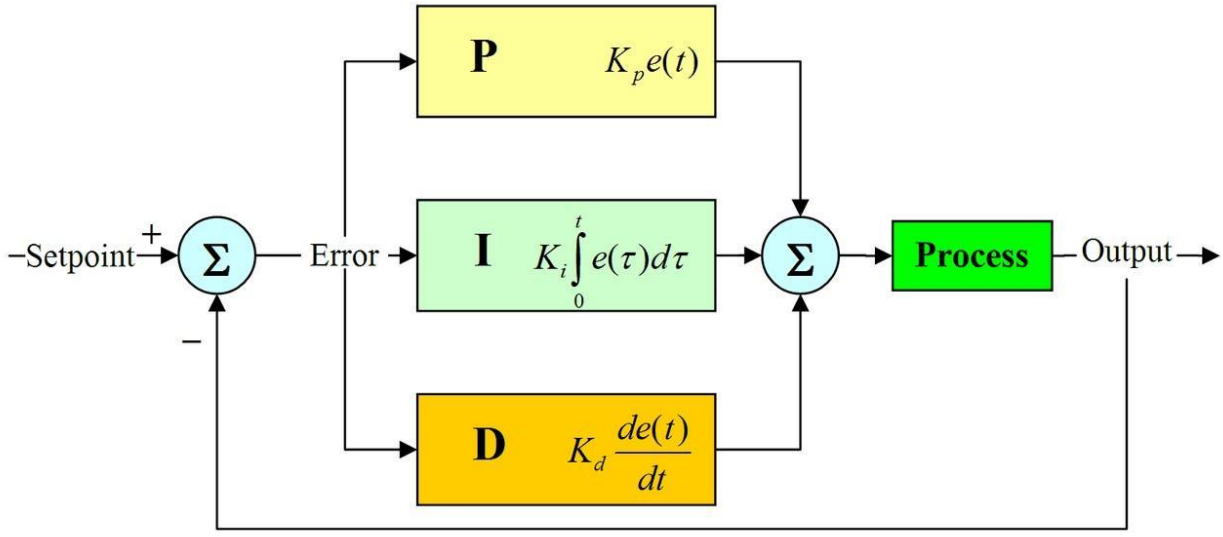


Figure [2-10]

2.3 Hardware Components

The vehicle requires a set of specialized sensors to help in various operations in the autonomous modules, the sensors used for the autonomous robot are GPS, IMU, potentiometer, accelerometer and ultrasonic sensor.

The GPS stands for global positioning system, this system uses the combination of responses from different satellites to estimate the position of the receiver on planet earth, using the longitude and latitude coordination system, this would help us in initially estimating the position of the robot globally on earth. [17]

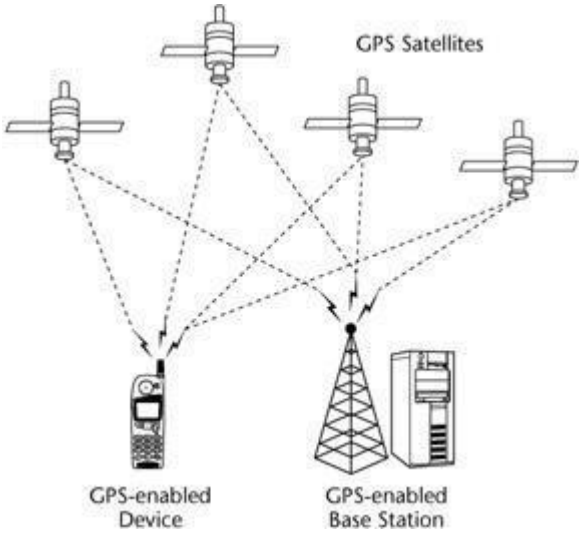


Figure [2-11]

The IMU stands for inertial measure unit, it depends on a magnetic gyroscope which outputs the rotations in the three directions, which are the yaw, pitch and roll, this helps us in determining the robot's current rotation and which direction it is facing at any moment. [18]

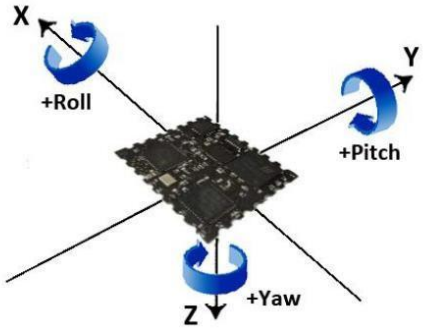


Figure [2-12]

potentiometer is a type of position sensor. They are used to measure displacement in any direction. Linear potentiometers linearly measure displacement and rotary potentiometers measure rotational displacement.

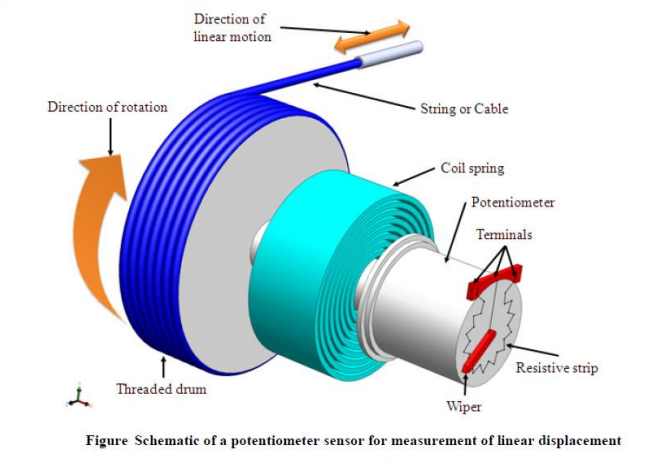


Figure Schematic of a potentiometer sensor for measurement of linear displacement

Figure [2-13]

Accelerometer is a sensor used to detect the change in the velocity in the three main axes ‘X, Y, Z’, this will be used to estimate the velocity of the car, which will help us in estimating its position and in adjusting the outputs of the controller. [19]

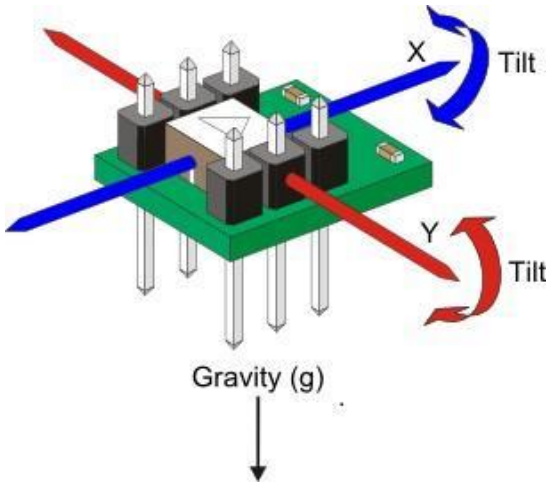


Figure [2-14]

The ultrasonic sensor's main idea is to send a wave and receive it after a specific time, knowing its speed we can estimate the distance between the source and the object that the beam was reflected from. [20]

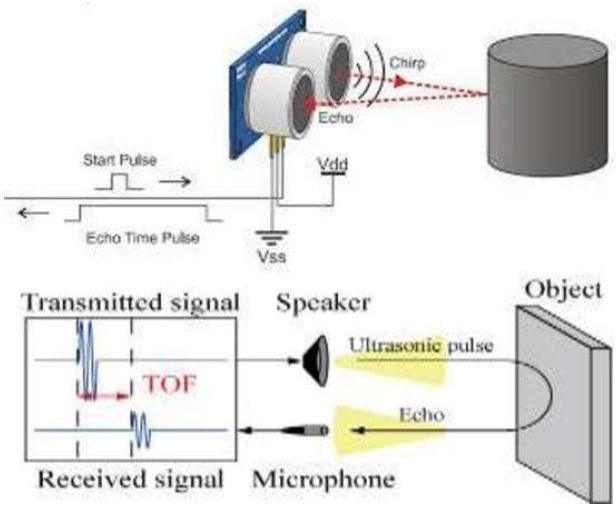
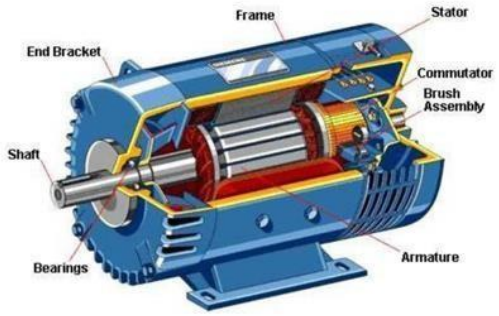


Figure [2-15]

A DC motor is any of a class of rotary electrical motors that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current in part of the motor.

DC motors were the first form of motor widely used, as they could be powered from existing direct-current lighting power distribution systems. A DC motor's speed can be controlled over a wide range, using either a variable supply voltage or by changing the strength of current in its field windings. Small DC motors are used in tools, toys, and appliances. [21]

The universal motor can operate on direct current but is a lightweight brushed motor used for portable power tools and appliances. Larger DC motors are currently used in propulsion of electric vehicles, elevators, and hoists, and in drives for steel rolling mills. The advent of power electronics has made replacement of DC motors with AC motors possible in many applications.



DC Motor Breakdown

Figure [2-16]

An H-bridge is an electronic circuit that switches the polarity of a voltage applied to a load. These circuits are often used in robotics and other applications to allow DC motors to run forwards or backwards.

Most DC-to-AC converters (power inverters), most AC/AC converters, the DC-toDC push–pull converter, isolated DC-to-DC converter, most motor controllers, and many other kinds of power electronics use H bridges. In particular, a bipolar stepper motor is almost always driven by a motor controller containing two H bridges.

H-bridges are available as integrated circuits or can be built from discrete components. [22]

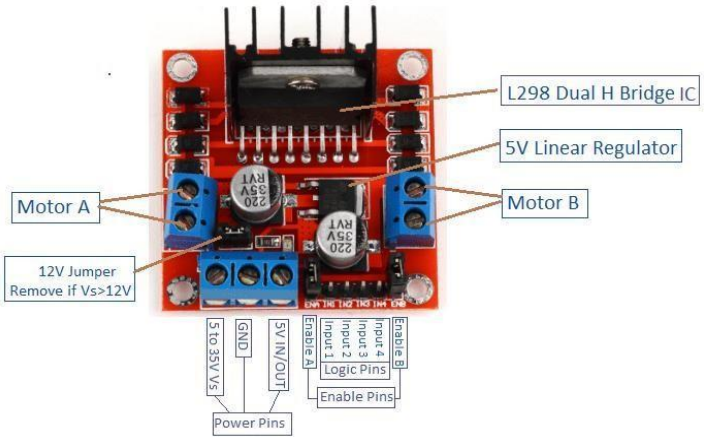


Figure [2-17]

The term H-bridge is derived from the typical graphical representation of such a circuit. An H-bridge is built with four switches (solid-state or mechanical). When the switches S1 and S4 (according to the first figure) are closed (and S2 and S3 are open) a positive voltage is applied across the motor. By opening S1 and S4 switches and closing S2 and S3 switches, this voltage is reversed, allowing reverse operation of the motor.

Using the nomenclature above, the switches S1 and S2 should never be closed at the same time, as this would cause a short circuit on the input voltage source. The same applies to the switches S3 and S4. This condition is known as shoot-through.

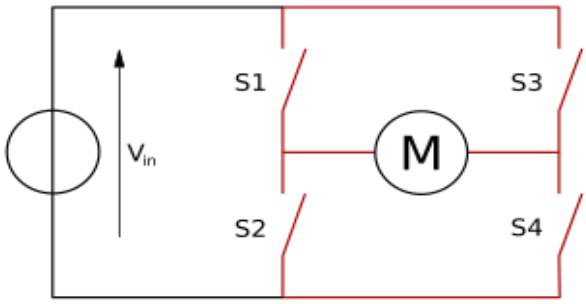


Figure [2-18]

Chapter Three
System Architecture

System Architecture

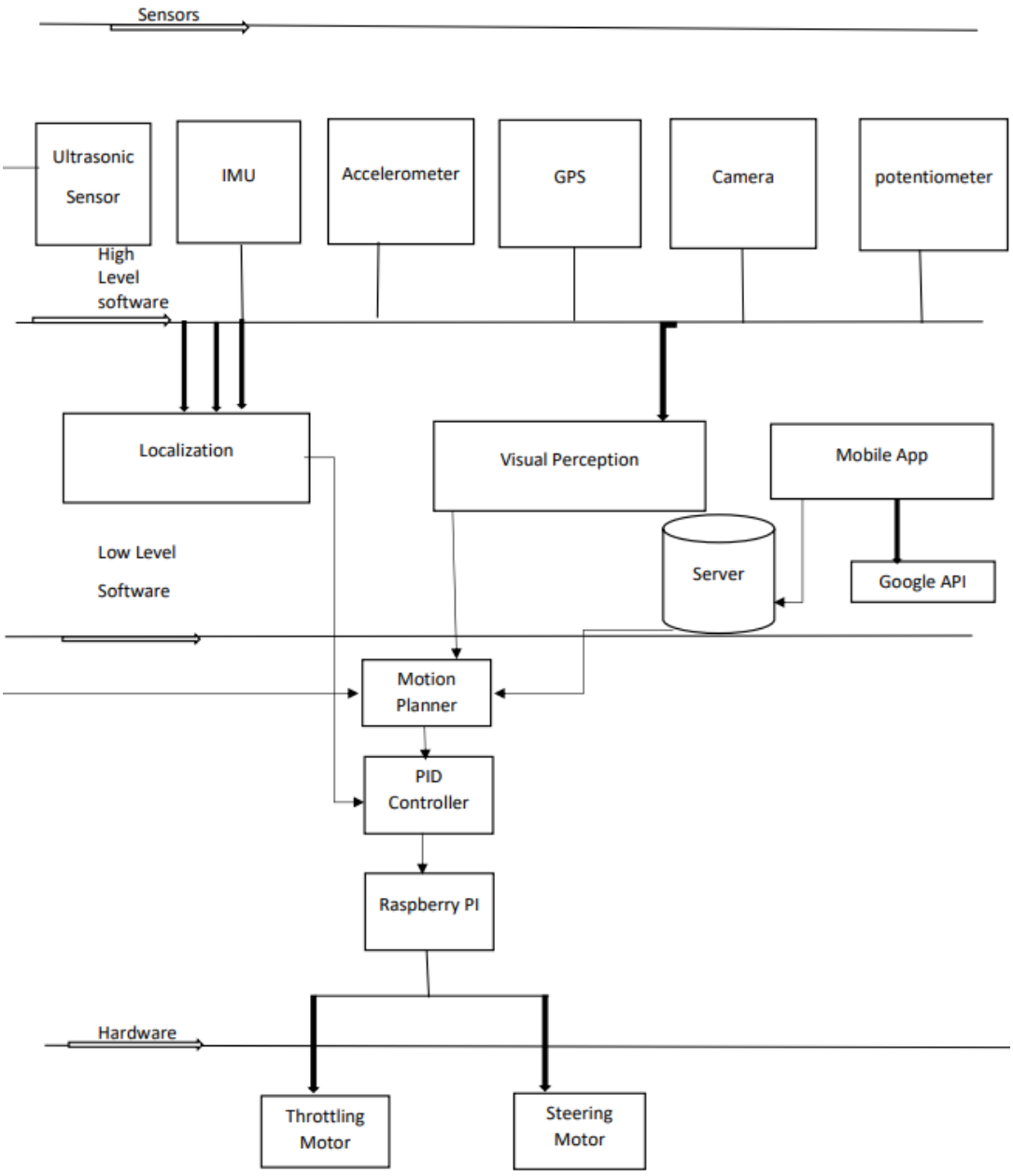


Figure [3-1]

The system consists of many interacting modules, some are dependent, and some are independent, it is considered a complex modulus relation as some modules are hierarchically on each other, and distributed computing is applied so some raw data are taken from different devices to be processed on more computationally efficient devices and then the results are given back to the source or to a new device.

The System architecture starts from the raw data which are taken from the input sensors, which are the camera, ultrasonic sensor, IMU, accelerometer,GPS and the potentiometer, each send its own raw data, to the devices to be processed and turn into useful information for the system.

As for the second layer the intensive computations are at their peak, so the system passes the high computational tasks to a server cluster which processes the data and returns the useful information, this happens with the camera input being passed to compute the visual perception part along with the app.

For the localization part we used the IMU, GPS ,accelerometer and the potentiometer sensors raw data then it computes the required filters to process the raw data and then pass it to the motion planner, after that the motion planner passes its data to the controller which in order convert all the software instructions into voltage values and passes them to hardware parts.

Mobile application was implemented, so that through it the client can make his orders and the destination will be sent to the car to follow and then qr-code will be sent to the client so he can scan it on the car and get his order , server save all data get from mobile app to can use it .

The whole communication phases are implemented using a local wireless network, as the robot should not be attached physically to all the devices to ensure the flexibility in movement and navigation.

Chapter Four
Software System Implementation

4.1 Robot Autonomous System Implementation

4.1.1 Autonomous Architecture Implementation

Combining the autonomous system architecture components introduces us an autonomous vehicle with level four autonomous level, in figure[4-1] it illustrates the process in which the system works, as it starts with combining the sensors data as in the sensor fusion part, then passing it to the localization module which estimates the position of the vehicle, then the visual perception which predict and estimate the behavior of the surrounding vehicles passing it to the motion planner which creates the path which the vehicle would follow, finally everything would be passed to the controller which gives to orders to the motors to move.

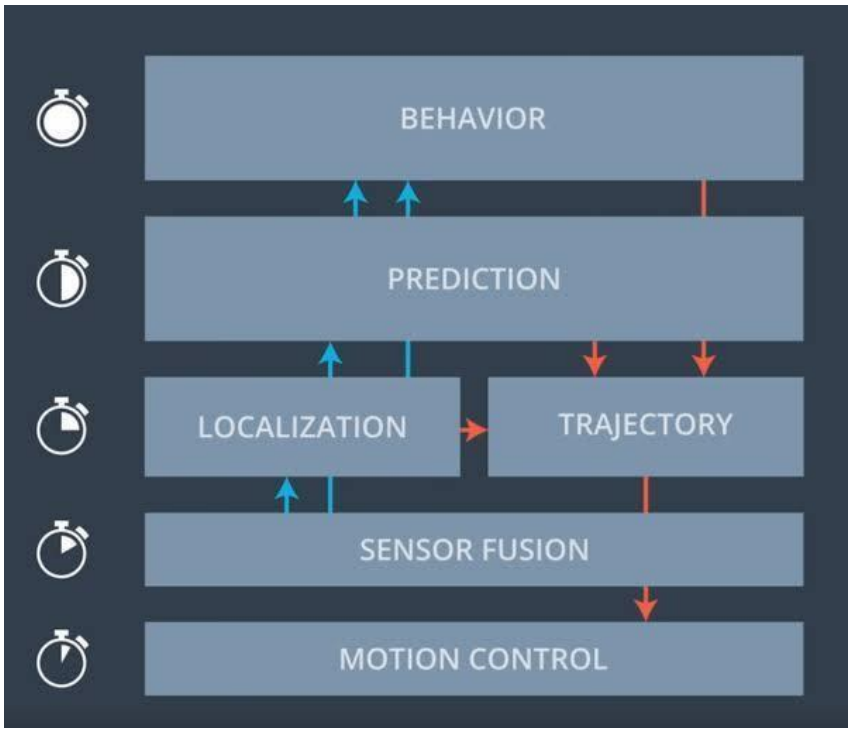


Figure [4-1]

4.1.2 Sensors

We did pin two ultrasonic sensors in the robot's front, to be able to sense the obstacles in front of the car while in longitudinal movement, and the two rear sensors are for sensing the incoming obstacles while performing a lateral action.

Figure [4-2]

We used Neo-8mm GPS Module to estimate the current position of the robot and to reach the desired location, GPS is also used as a Navigation System with Time and Ranging.

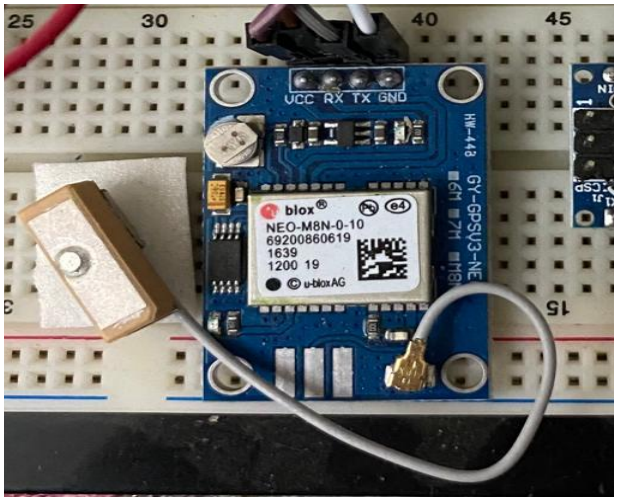


Figure [4-3]

We used the MPU6050 sensor module, it combines a 3-axis Gyroscope, 3-axis Accelerometer and Digital Motion Processor all in a small package. Also, it has an additional feature of on-chip Temperature sensor, It can be used to detect angle of tilt or inclination along the X, Y and Z axes ,Acceleration along the axes deflects the movable mass.

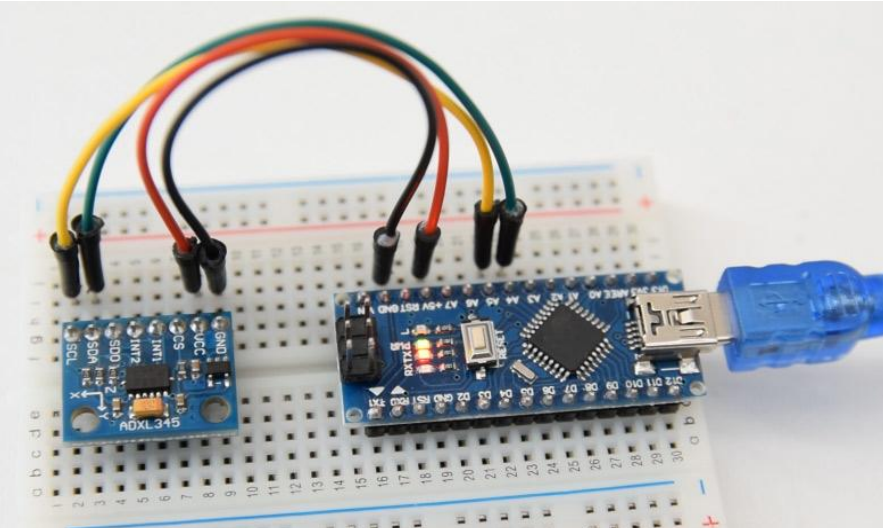


Figure [4-4]

We used a potentiometer sensor to measure the displacement of the robot in a linear or rotary motion and convert it into an electrical signal.

Figure [4-5]

We took the output of the four listed above sensors and passed them to the localization module in order to estimate the current position of the robot, velocity and the yaw angle.

4.1.3 Robot's Localization

After reading the sensors from the mobile and getting the current velocity from the accelerometer and the location from the GPS and the yaw angle from the gyroscope in the IMU sensor unit, we apply Kalman filter to them to compute the most accurate position of the robot, by getting the GPS readings then uses the IMU and accelerometer outputs to try to estimate the current location, and update the values for each new GPS signal.

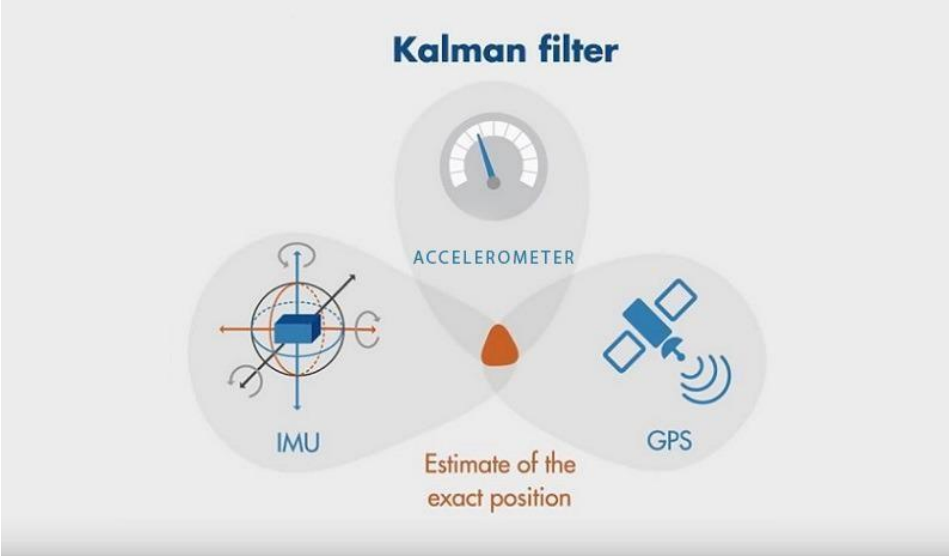


Figure [4-6]

4.1.4 Visual Perception

In order to achieve the visual perception, we need to get information of the scene around the robot, so we first we needed to use an object detection model to achieve this task, so we used YOLOV4 object detection to detect the objects in front of the robot to be able to predict the behavior of the world around then analyze it to make it able to choose the best decision to perform to achieve a successful navigation during its mission.

After understanding the environment's objects, we need to determine the positions and the locations of them relative to each other, also we need to identify where the road is and its curvature, so we needed to use a semantic Lane detection model to achieve this.

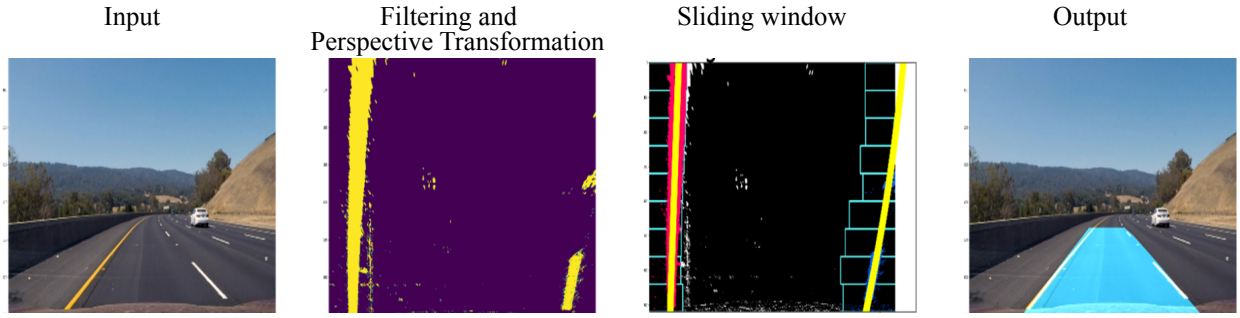


Figure [4-7]

4.1.5 Motion Planner

The motion planner module mission is to determine the exact points that the robot must follow. We implemented this module by taking the input sensors of the ultrasonic to detect the existence of obstacles in front of the robot.

We also used the input data from the visual perception which determines the objects in the scene and predict their movement, which allows us to give a good action for the robot to perform, which is in a nutshell updating waypoints for the robot.

We used google maps as a map provider to initially get a set of longitude and latitude set of way points of the customized journey that we wish the robot to travel, after that we convert the route extension into GPX extension to be able to extract the set of way points directly. [23]

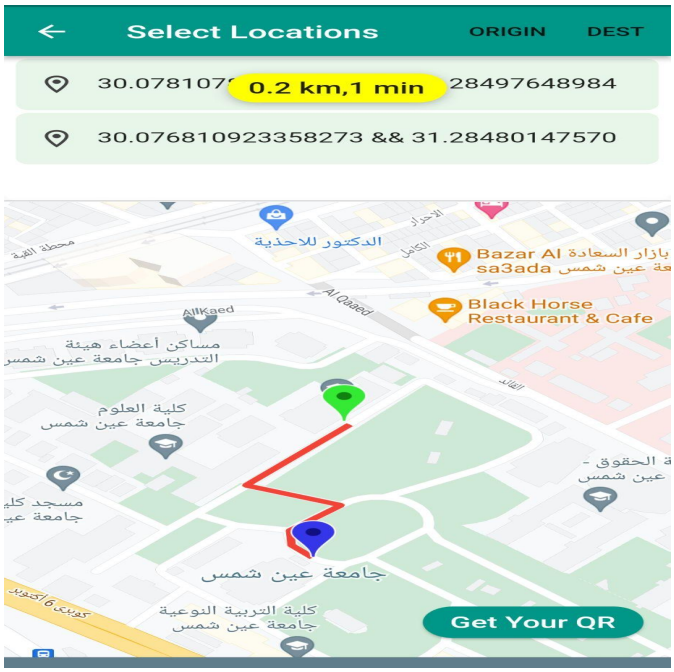


Figure [4-8]

After getting the way points generated from our maps provider, we have to give it an extra layer of filtering as the GPS points are general and distant, on an actual road it's not enough for the robot to follow them, as shown in the below figure, consider the AB line as the actual road and the red line as the GPS points, we would notice that the robot will follow points X1, X2, X3, X4, X5, as expected the robot will move on a curvy road and the navigation stability would be minimized.

So, the solution for this was interpolating the line between the robot's current position and next five waypoints ahead of it, and creating a virtual straight line from the robot's position to the average of the taken GPS samples, creating an estimation of the road waypoints, after that we would fit the sampled points with the virtual straight line creating the points Y1, Y2, Y3, Y4 ,Y5, which are the corrected versions of the generated way points, and that would produce a more stable navigation for the robot

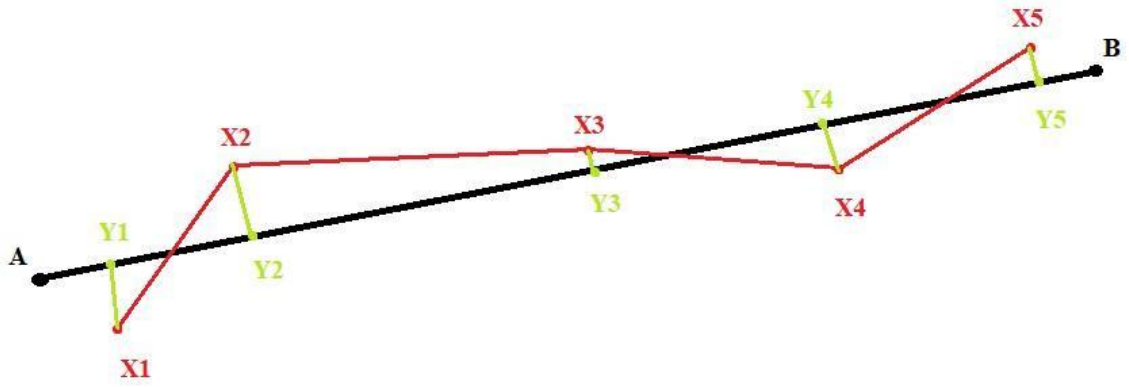


Figure [4-9]

Adjusting the straight line problem, we faced the curves problem, as applying the same algorithm which is assuming that the road ahead the robot is a straight line would produce failures, so the procedure we took for solving this problem is first detecting if the sampled points that we took more likely form a curve or a straight line, so we would pass the sampling points to a straight line equation and check the distance between each point and the nearest point on the line to it, then we would pass the points with a quadratic function and check the distance between each point and the nearest point on the line, then we would measure the error using mean squared error algorithm. [24]

Then the equation with the least error will indicate whether the following set of lines form a curve or a straight line, if they form a straight line, the algorithm mentioned above will be used, else we would use a filter for the curve case.

To filter a curved path, we will use the generated equation to try to fit the random points same as we did with the straight line problem, the start of the parabola will be the robot's current position, as we did before points Y1,Y2,Y3,Y4,Y5 are the corrected path generated from the points X1,X2,X3,X4,X5.

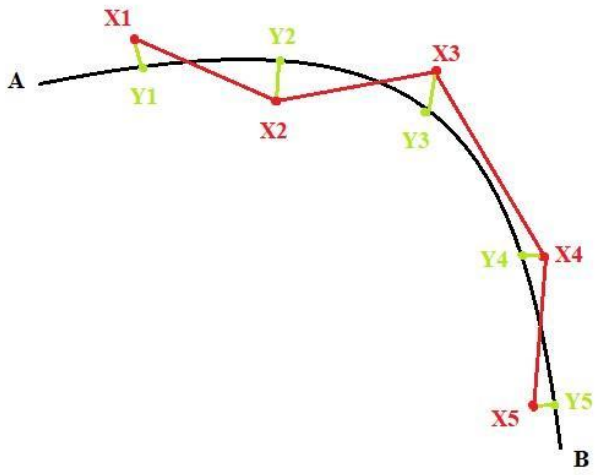


Figure [4-10]

After getting the filtered waypoints that the robot will follow, we proceed with another level of filtering, as this time we will use the information gained from the visual perception to update and correct the waypoints and the ultrasonic sensor which determines if an obstacle is ahead or interfering the waypoints, this will result in a behavior of the two.

whether the robot would stop allowing the interfering object to move in case it is a dynamic object like pedestrians or cars, or it would move around the obstacle if it were a static object like a boulder or a blockade.

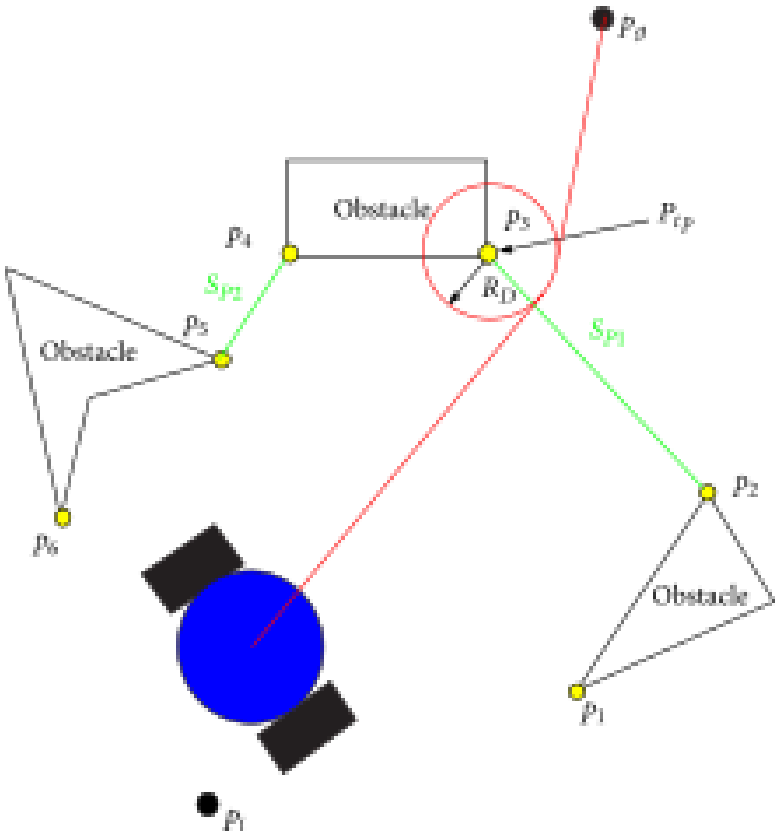


Figure [4-11]

So, this will generate a form of a finite state machine, where the robot will move and follow its waypoints until an event occurs which will lead the robot to stop or to update its route resulting in the robot changing its route. [25]

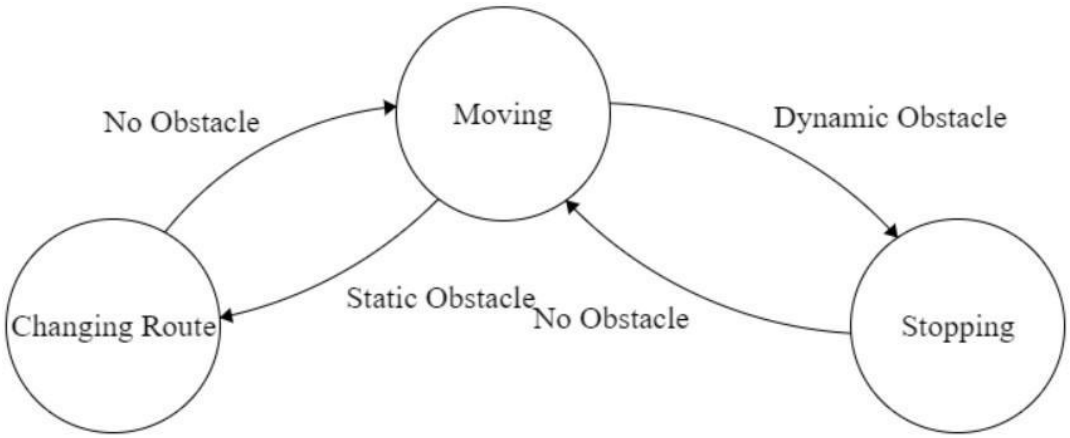


Figure [4-12]

By applying this finite state machine, the robot's movements and decisions are well bounded and studied so that computing the best decision by processing the incoming data, filtering the waypoints, and understanding the scene from the visual perception also giving in consideration the distances between the robot and the objects is well achieved generating a smooth control and movement for the robot.

4.1.6 PID Controller

We implemented the PID control loop to get the required motion values for the throttling values and the steering values to pass it to the motors so that we achieve the desired position that we wish our robot to follow.

So, in our case the inputs for the PID controller will be the robot's current velocity, desired velocity, current position, the desired position, the current yaw, and outputs the throttle value, which ranges from 0 to 1, this value will feed the robot's motors

as they will represent the value of the percentage of the voltages we want to pass to the motor, which determines the velocity of the robot we desire it to move with.

As for the steering, the PID controller also outputs the desired angle it wishes to rotate the car into, it outputs the value ranging from -1.2, to 1.2 in radian, and this value is taken and processed so we can give the steering motor a certain percentage of voltage for it to operate.

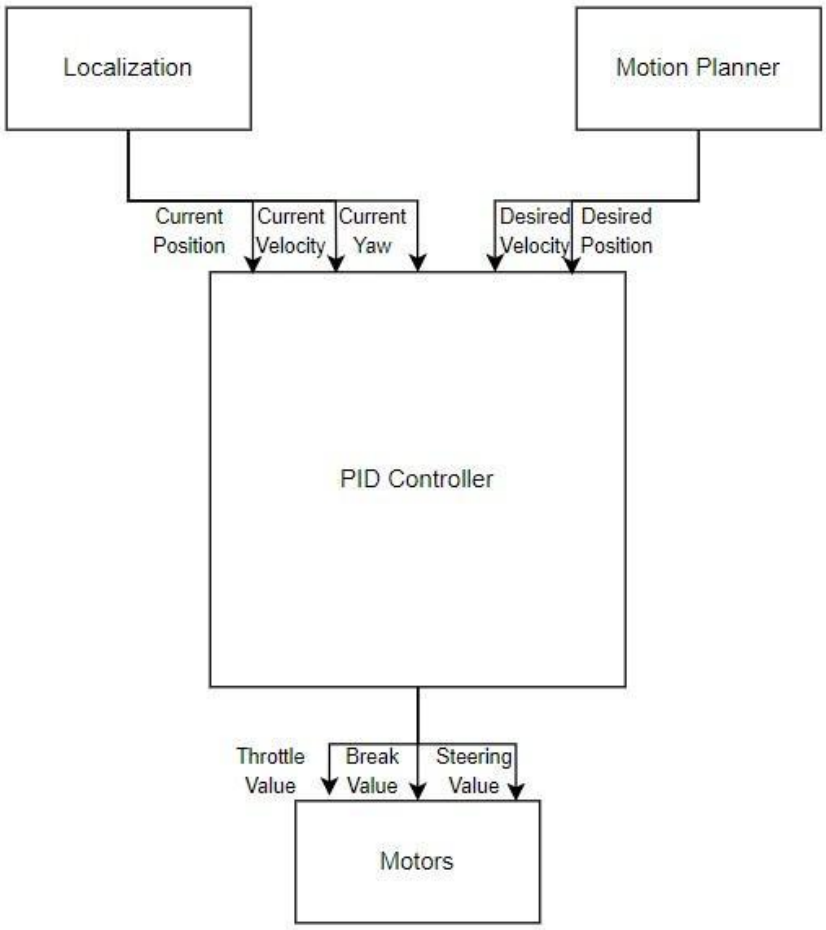
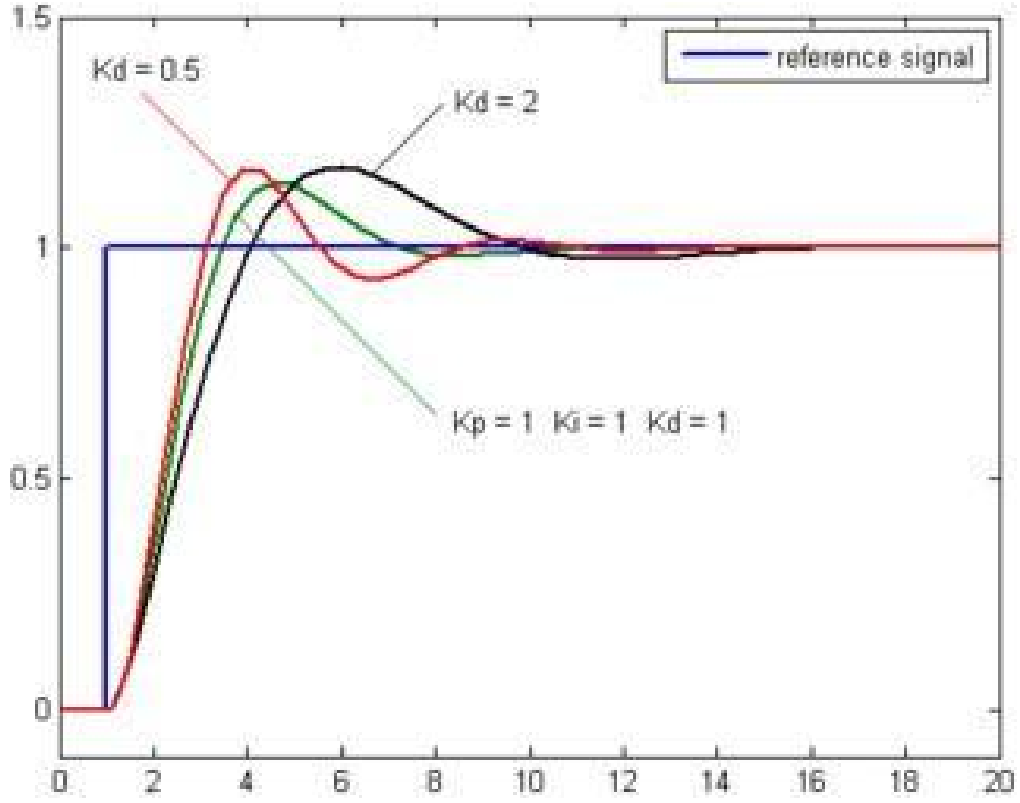


Figure [4-13]

After implementing our PID we faced the problem of tuning the PID parameters, so we tried following our controller's behavior in order to tune the PID parameters and achieve a good state for our robot's navigation through the given route.


reference signal
Kd = 0.5
Kd = 2
Kp = 1 Ki = 1 Kd = 1
1.5
1
0.5
0
0
2
4
6
8
10
12
14
16
18
20
Figure [4-14]

4.2 Implementation of Visual Perception System Modules

4.2.1 Visual Perception System Architecture:

The visual perception architecture depends on receiving the camera frame, then passing it to four running deep learning models to extract the information and process the raw data, to understand the existing environment around the car.
The series connected models are the traffic signs and lights detection and traffic signs and lights recognition, and object detection and object recognition, as running them in parallel would be a waste of resources, that's because if there are no signs or objects found, there won't be a sign or object to recognize.
So, performing detection first saves a lot of computations.

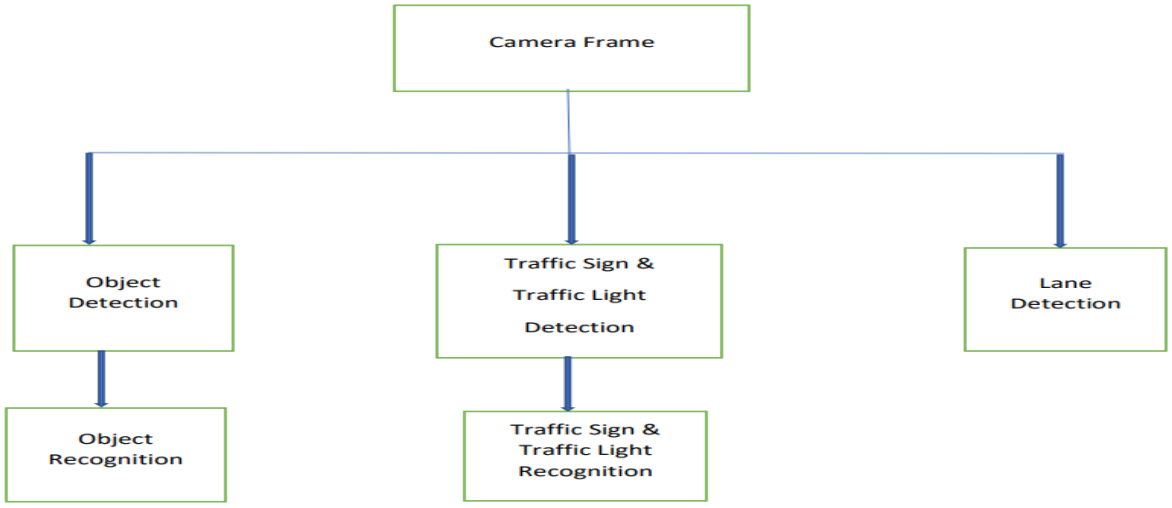


Figure [4-15]

4.2.2 Object Detection

Object detection and recognition is a very important phase, as through it we can understand the environment around.

We implemented an object detection model using YOLO V5 architecture to detect and recognise different objects. The car will use the object detection system to recognize and understand the surroundings it encounters during its driving to make the right decisions. so, for example, it can understand when there is a human in front of it or a car so it can adjust its speed or even stop driving.

YOLO is an algorithm that uses neural networks to provide real-time object detection. This algorithm is popular because of its speed and accuracy.

First, the image is divided into many grids. Each grid has a dimension of M x M. Each grid cell forecasts B bounding boxes and provides their confidence scores.The cells predict the class probabilities to establish the class of each object. Then Intersection over union is applied to provide an output box that surrounds the objects perfectly. The IOU will equal to 1 when the predicted bounding box is the same as the real box

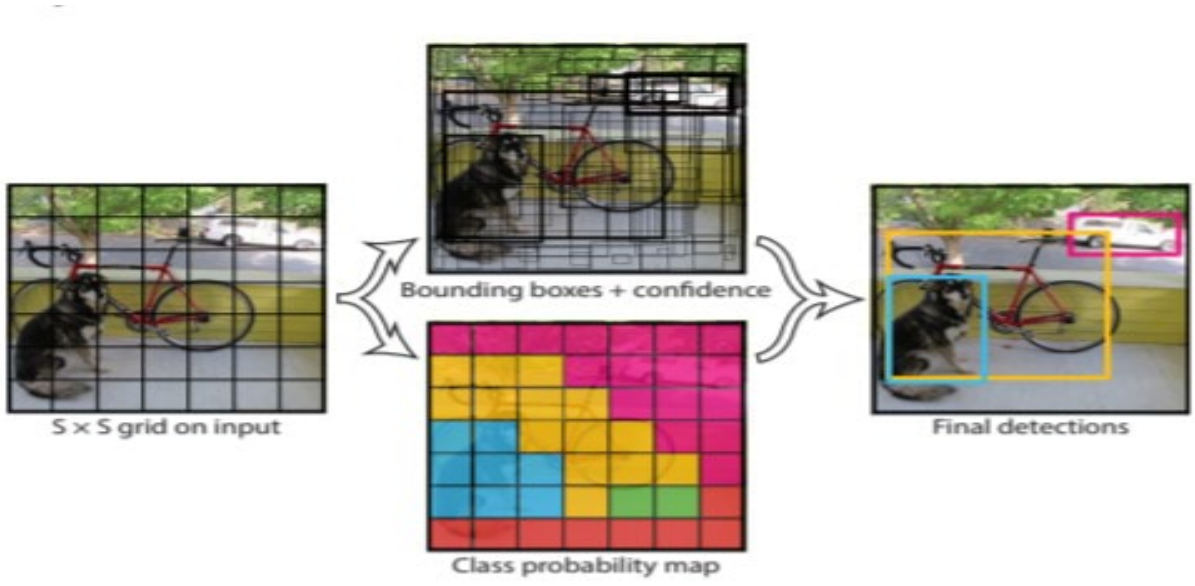


Figure [4-16]

The model was trained on the COCO dataset deployed using a tiny yolo model by only using one frame to train the model and extract the location of the plate in frame.

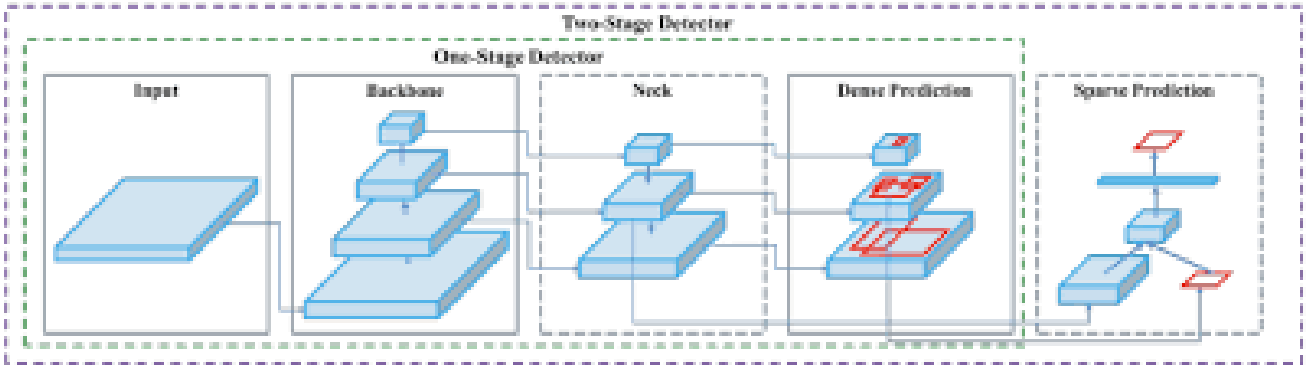


Figure [4-17]

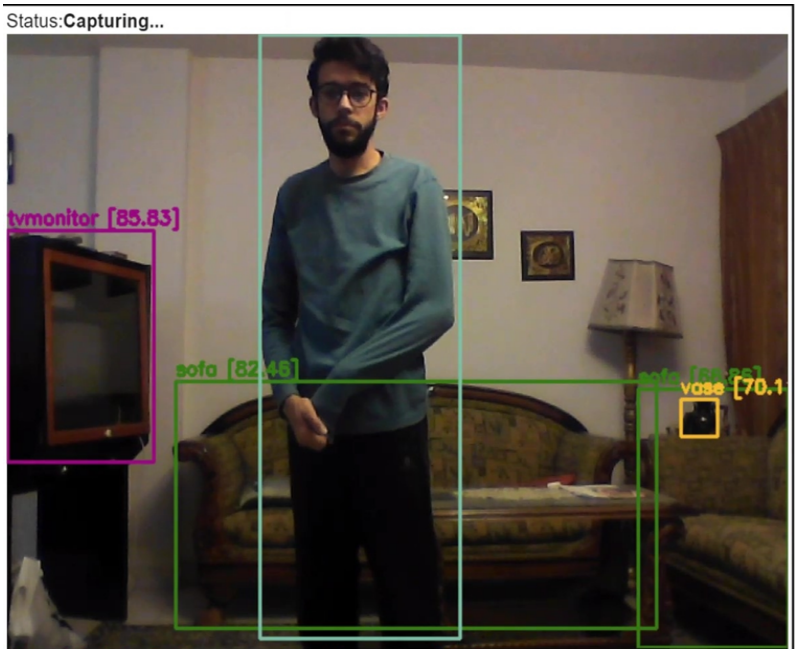


Figure [4-18]

4.2.3 Traffic Signs and Lights Detection

We implemented a traffic signs and lights model using YOLO V4 architecture to detect the traffic signs and lights, then crop them from the image and apply recognition on them so the car can be able to differentiate between the different types of signs and the different colors of lights.

Detection phase

A model was trained on COCO dataset using YOLO v4 so it can detect the traffic signs and lights.

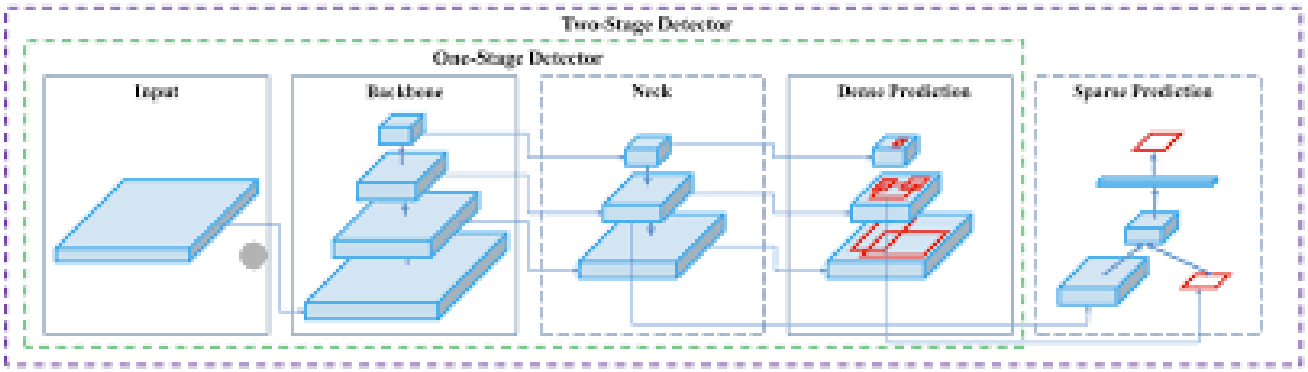
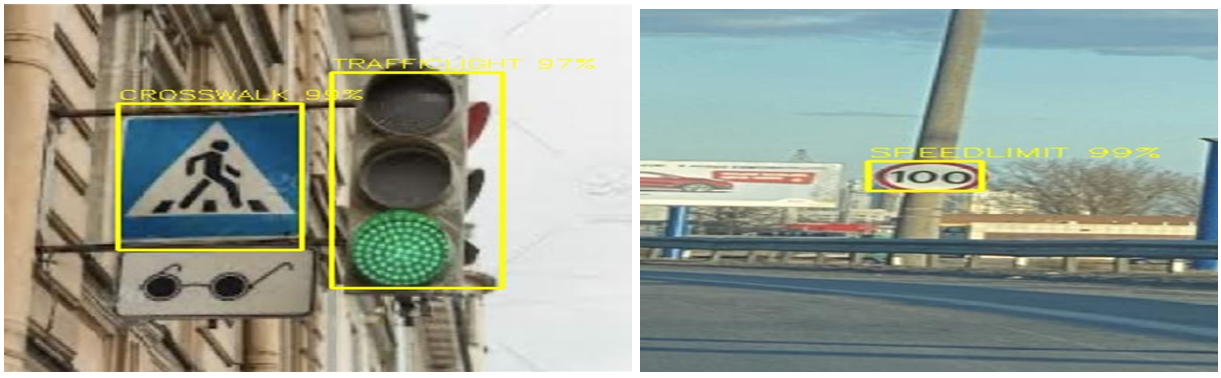


Figure [4-19]



. Figure [4-20]

Recognition Phase

After detecting a speed-limit traffic sign or a traffic light we then need to recognize what it is.

We first cropped the region of interest.

Figure [4-21]

And then we used image processing techniques (Image thresholding, Pytesseract) to classify the traffic light color and the speed limit number.

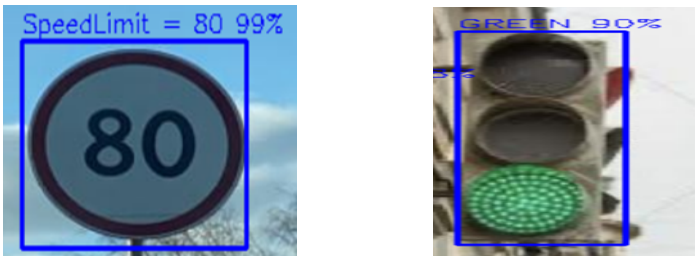


Figure [4-22]

Final outputs

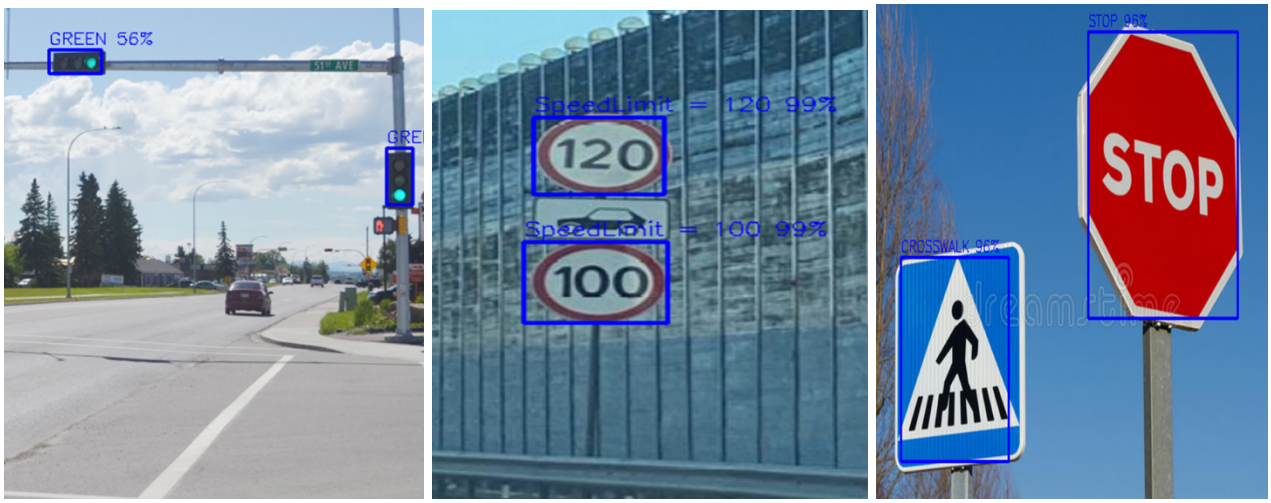


Figure [4-23]

4.2.4 Lane Detection

Detecting the road lanes so the car can be able to define its boundaries. Calculating the lane curvature and the vehicle offset to help with instructing the car for the right driving and to help maintain the car in its own lane without sliding away from it.

To build a good lane-detection model we made the following points:

- Camera Calibration:
  To correct the distortion of the image of the scene
- Filtering and Perspective Transformation:
  To mainly focus on the driving ground with the lanes

. Figure [4-24]

- Sliding Windows and Curve Fitting
  To fit curved lines through the road lanes

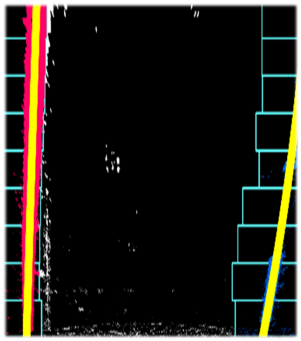


Figure [4-25]

combining all the previous steps we could finally detect the road lanes

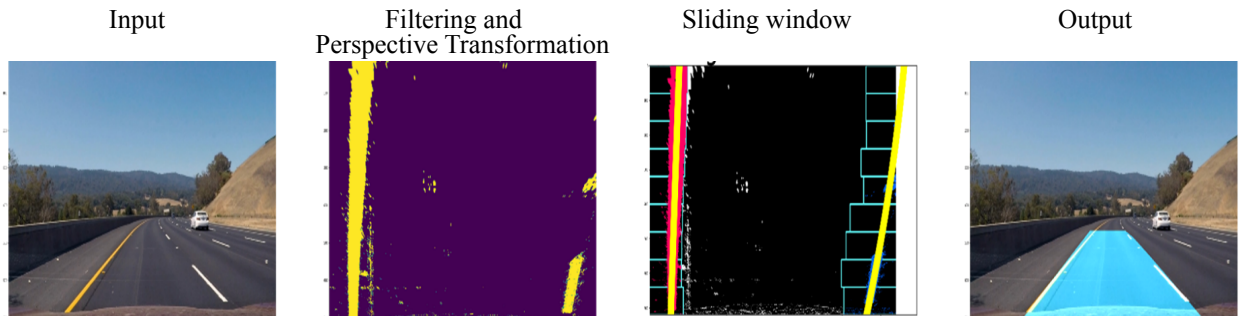


Figure [4-26]

4.3 Implementation of Smart Delivery Application

4.3.1 Delivery Application Architecture

Users can sign in on the Application, Set the order, and their desired destination.

The application generates a QR-code for each order, and it will be sent for the clients so they can scan it on the car to collect their order

There will be an employee that picks the items and puts it in the car and updates the status of the existing items and the order.

The application generates the path that will be sent to the car through the raspberry pi to follow to drive to the client destination.

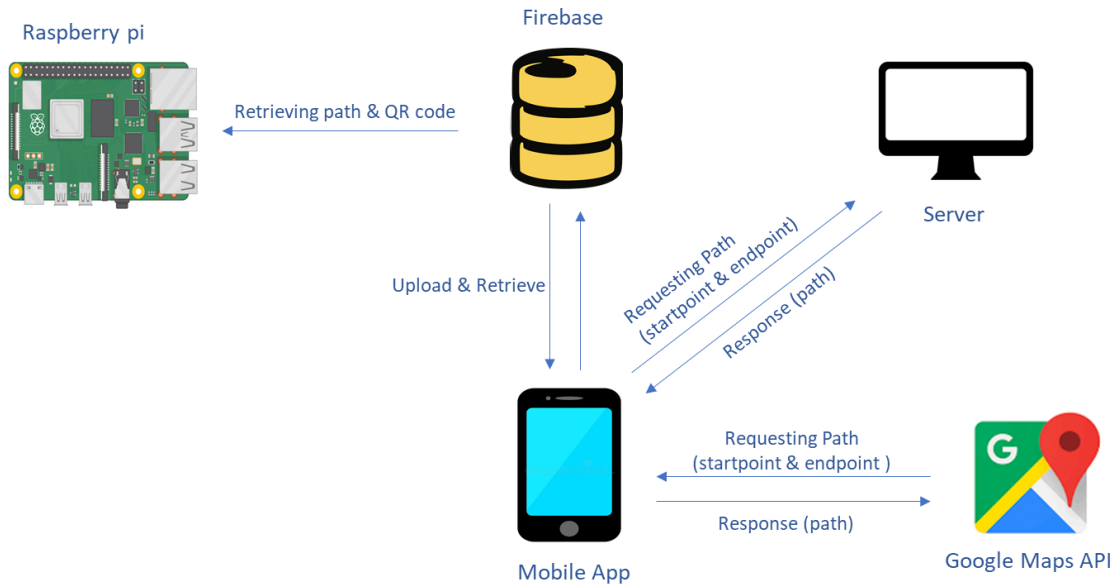


Figure [4-27]

4.3.2 Implementation of the delivery application

we use in our App:

- **Flutter**

  Flutter is an open source by Google for building beautiful, natively, compiled, multi-platform applications from a single codebase.

  Packages:

```
google_maps_flutter: ^2.0.2

location: ^4.1.1

flutter_google_places: ^0.3.0

flutter_polyline_points: ^1.0.0

qr_flutter: ^4.0.0

location: ^4.1.1
```

Figure [4-28]

- **Google Maps APIs**

  We have used Google maps APIs to get the desired paths with its longitude and latitude for any destination we want.

  APIs & Services:

```
Google Cloud APIs

Maps SDK for Android

Places API

Directions API
```

Figure [4-29]

- **Firebase**

  Firebase is an app development platform that helps you build and grow apps and games users love. Backed by Google and trusted by millions of businesses around the world.

Packages :

```
firebase_auth: ^2.0.0

firebase_core: ^1.3.0

firebase_storage: ^10.2.7

flutter_slidable: ^0.5.7

cloud_firestore: ^3.1.8
```

Figure [4-30]

- **Flask**

REST API services let you interact with the database by simply doing HTTP requests. In this article you learn how to write a REST server using Flask. This is often how the backend of web apps is created. Returning data is in JSON format and requests we are using are PUT, DELETE, POST, and GET.

Packages:

```
from flask.app import Flask

from flask import json, send_file ,request

from flask import Flask , request ,redirect ,url_for,render_template

from flask import jsonify

import base64
```

Figure [4-31]

we use Encode base64 to change the image to string and send the string to flutter app and in the app convert the string to image using decode

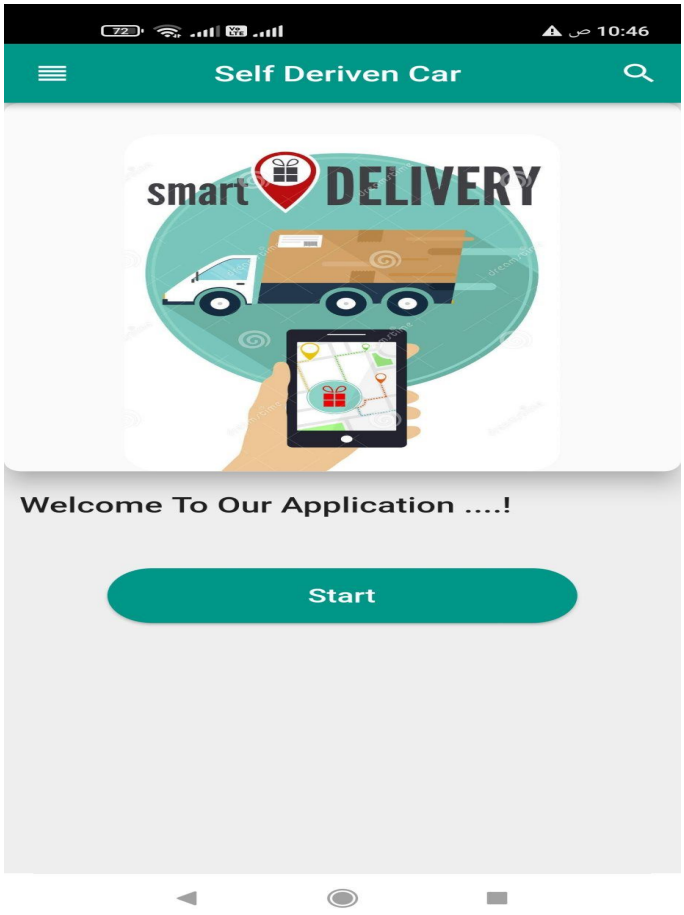
```
Uint8List bytesImage= Base64Decoder().convert(imagestring);

Image.memory(bytesImage),
```

Figure [4-32]

Screen illustration :

Welcome Page to our Smart Delivery Application



Figure[4-33]

Sign in and Sign up screen the user will enter his email and password to be able to login into the app and use it or can make a new account to be able to use the app.

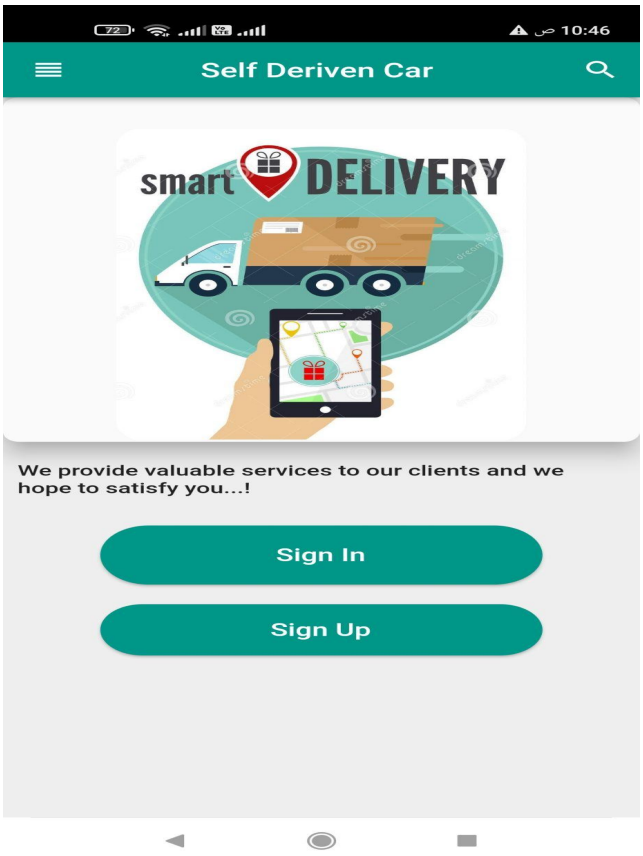


Figure [4–34]

New users can register a new account easily by entering his basic information with his password that will be used in the login method .

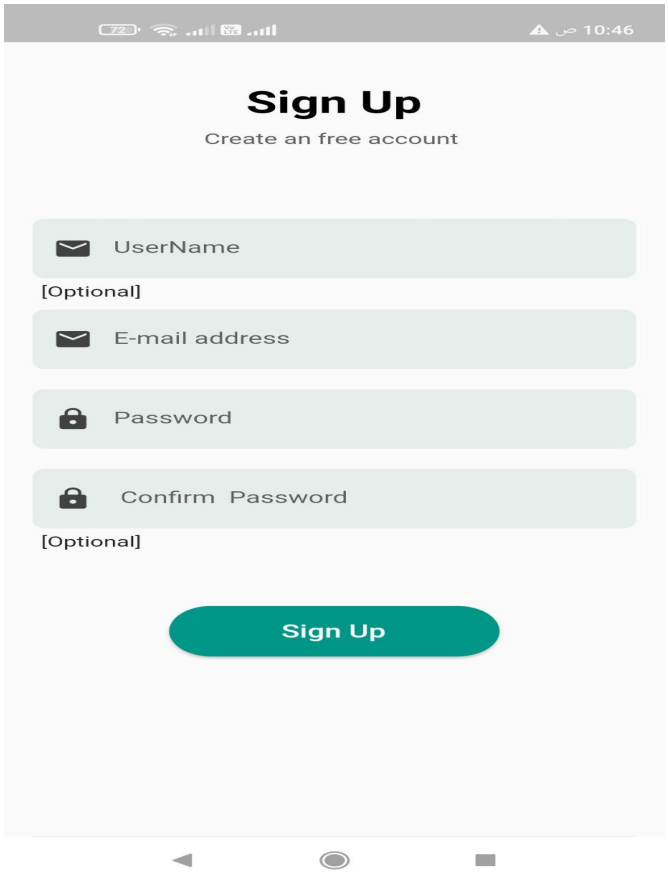


Figure [4–35]

After login client can make any order his want and set the quantity of the order then continue to ..

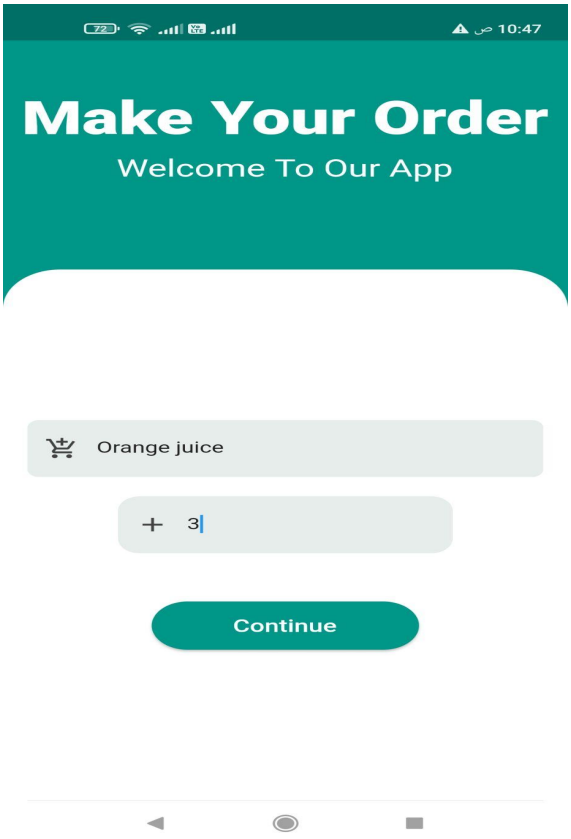
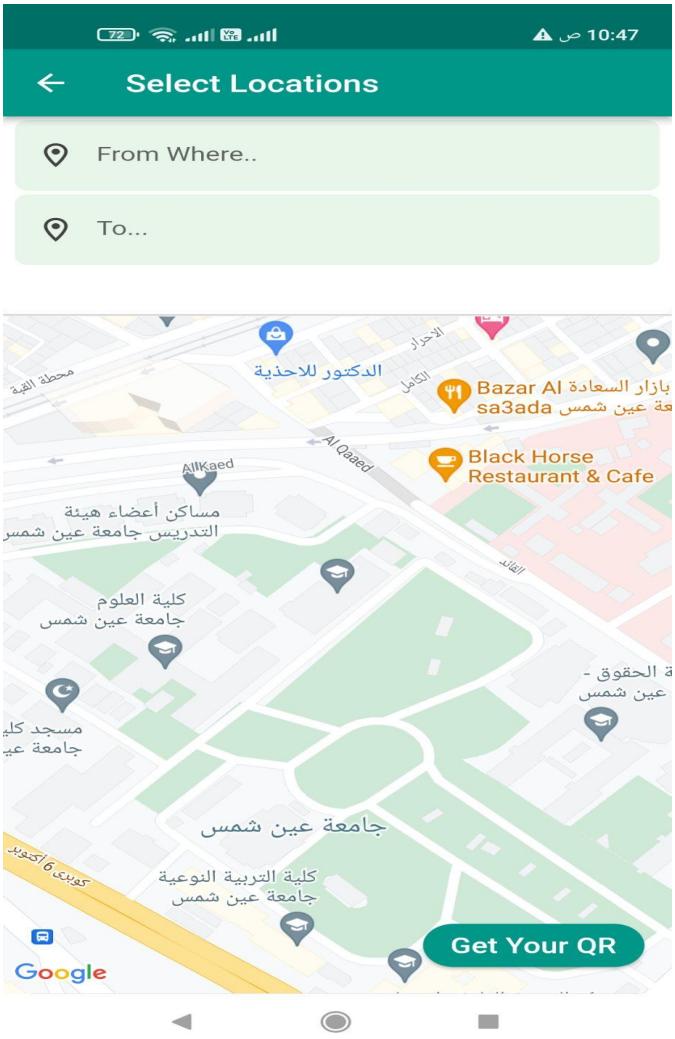


Figure [4-33]

Now the client can choose the Map he prefers between Global Map or Local Map .

Figure [4-37]

IF The Client chooses the Global Map,he can select the destination he wants the order to reach in it .



Figure[4-38]

If the Client Chooses the Local Map , he can select the destination he wants the order to reach in closed place,

Local Map use the server that created using Flask

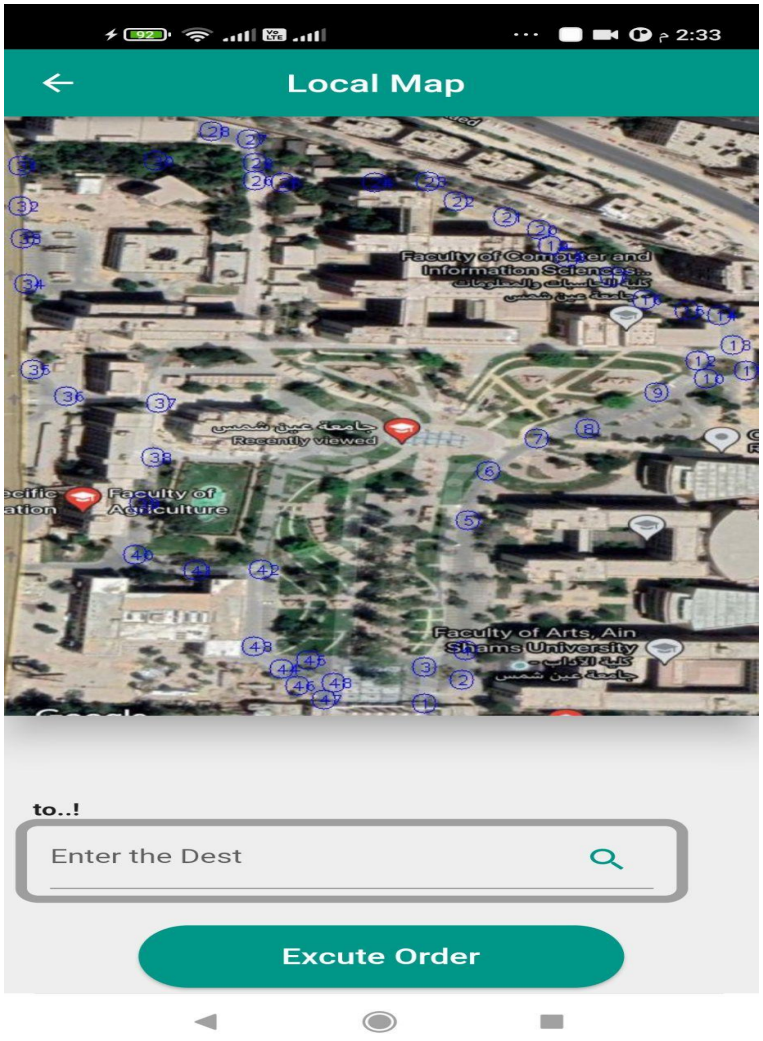


Figure [4-39]

When the client select the destination , the destination send to server to apply the algorithm to find shortest path (using A* Algorithm) then response the output in the app and save the path in the firebase

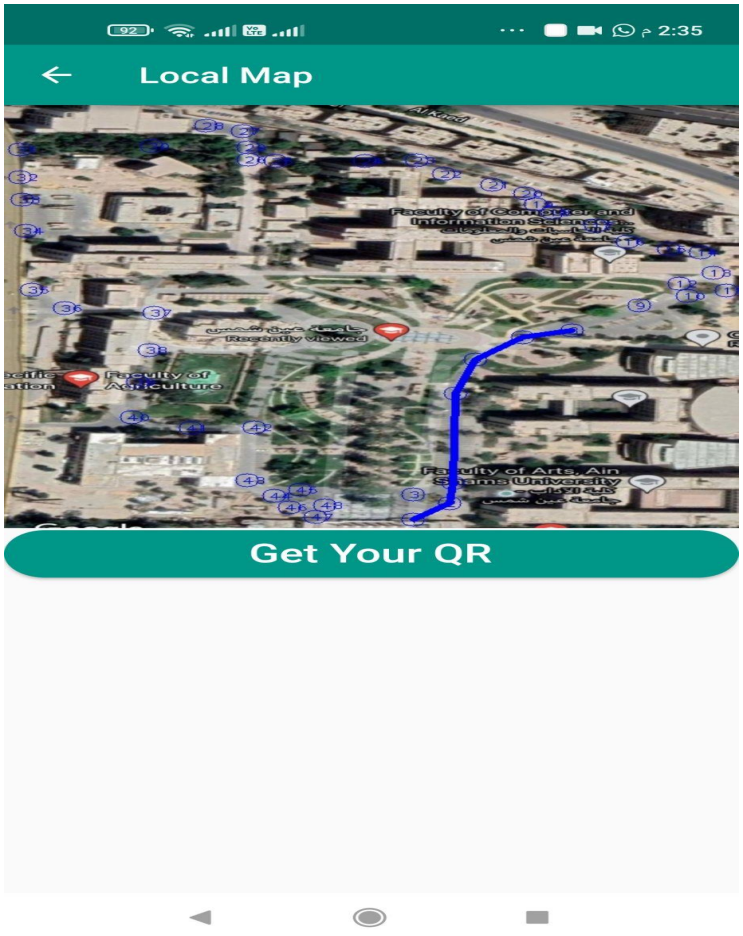


Figure [4-40]

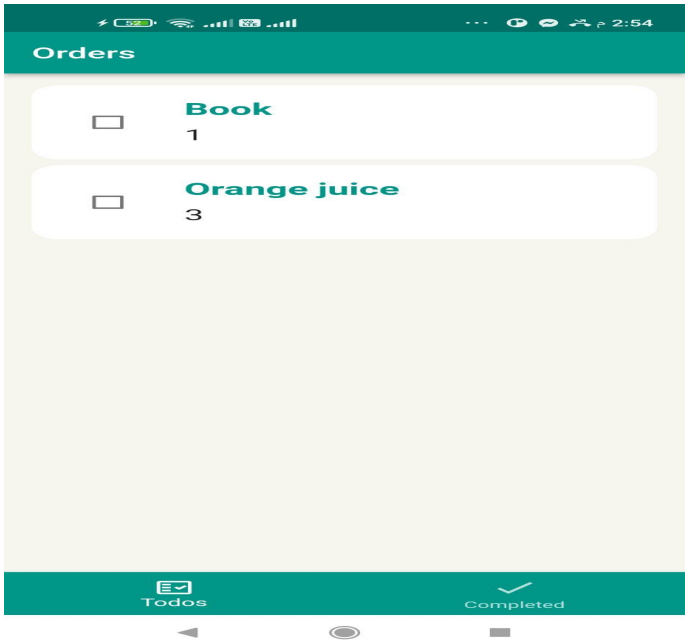
After setting his destination can get a qr-code to be able to receive his order and to verify that he is the desired client when the car reaches his place.

Figure [4-41]

IF the user is an employee , he can sign in and sign up to part of the employees in the app.

employees :

Orders:



Figure[4-42]

No Completed Orders:

Figure [4-43]

IF the employee update the status for any order that he picks it in the car:

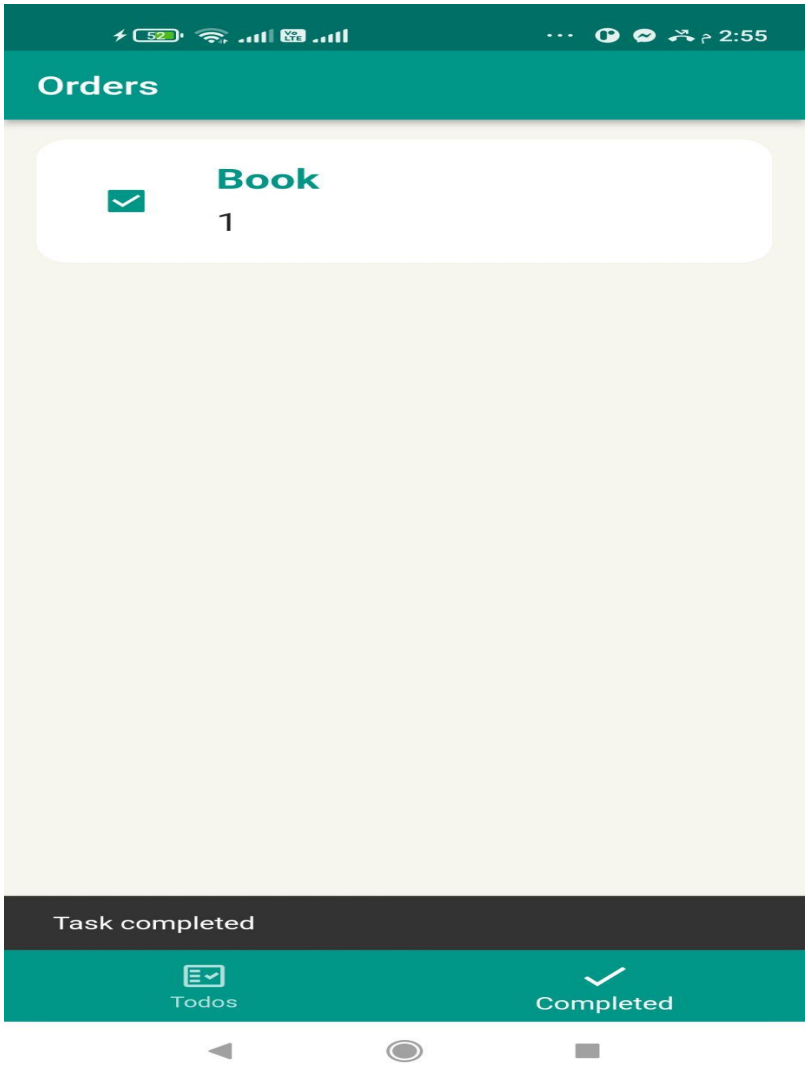


Figure [4-44]

Chapter Five
Hardware & Networking Implementation

5.1 Implementation of Robot's Hardware Structure

5.1.1 Data Devices:

Data devices are the devices used for data collection for the system, such as (Ultrasonic sensor, Camera, GPS, IMU, etc.).

Some of the data devices are connected directly to the micro-computer and some other devices cannot be connected directly with the micro-computer such as the Localization sensors that are embedded inside the mobile device.

Mobile devices are used to take advantage of the embedded localization sensors such as GPS and IMU Sensors.

Mobile Application is implemented to send those readings from the sensors to the micro-computer through sockets low level networking, to be able to use the mobile as data collector device the mobile phone must be installed on the moving car to update its localization data

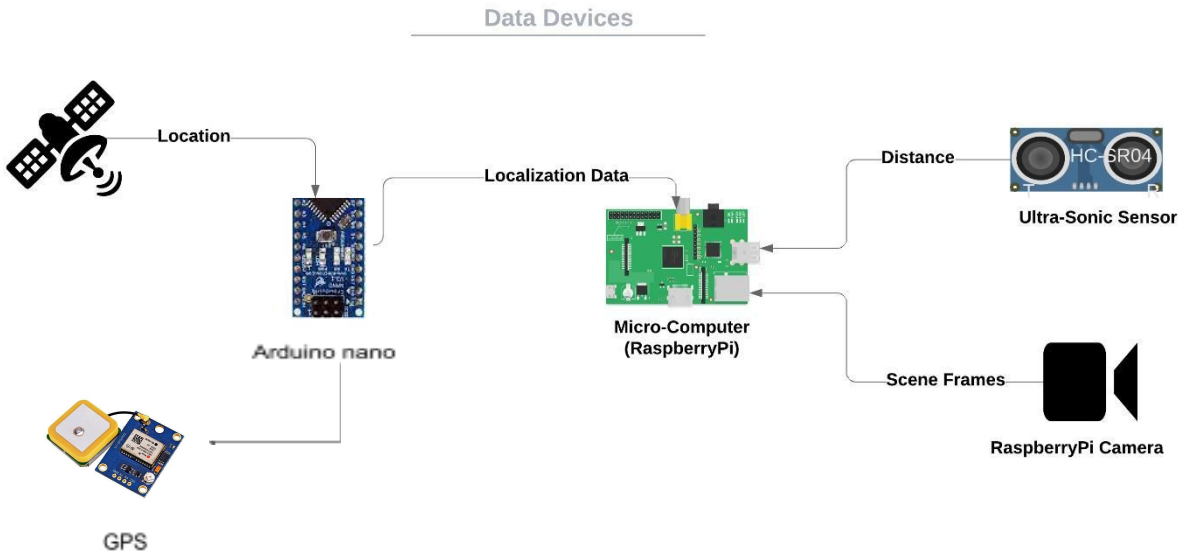


Figure [5-1]

5.2 Micro-Computer Setup

Micro-Computer is Printed circuit board of some components which forms small computer with a micro-processor as its central processing unit that may be connected with different Input/Output devices such as keyboard and mouse, also it may be connected with electronic devices and sensors such as Ultra-Sonic sensor, Camera module.

Micro-Computers contain Input/Output electronic pins used to control the electricity flow to control the motion of the patrol robot through a software.

5.2.1 Micro-Computers Version Selection:

Choosing the suitable type of Micro-Computer is very important since it acts as the central or main component of the system which collects data through sensors and the other system components then process these data and make motion decisions, so it requires good computational power.

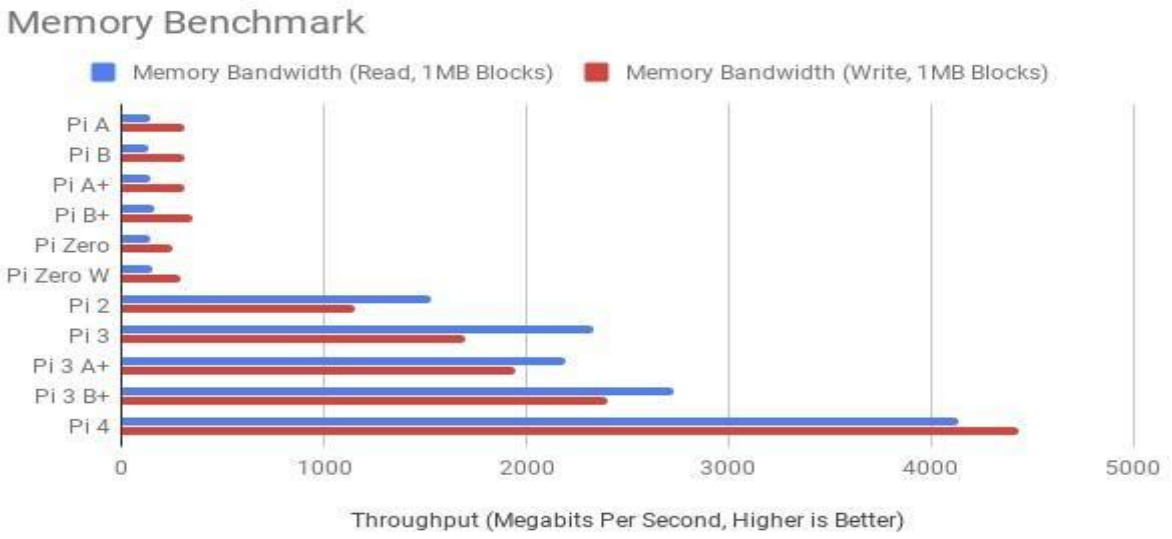


Figure [5-2]

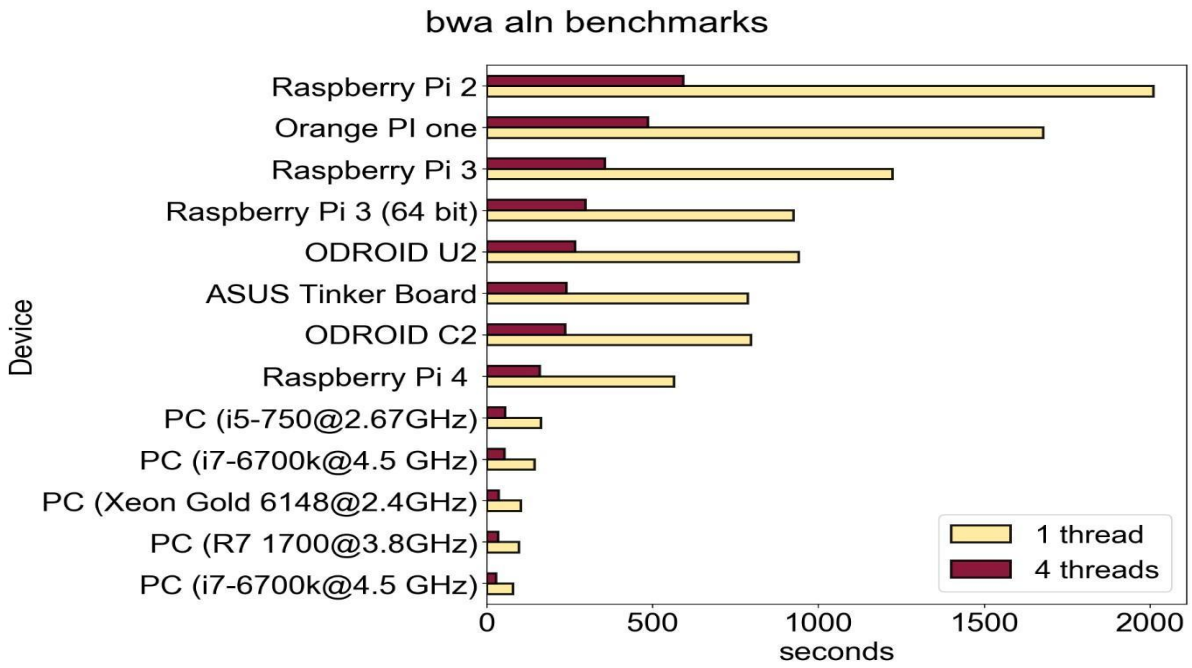


Figure [5-3]

According to the Latest to figures of benchmarks the Memory and the processing power of RaspberryPi 4 is the best micro-computer of this set of micro-computers

5.2.2 Environment:

RaspberryPi 4B Micro-Computer can run on an operating system that is dedicated to micro-computers such as Windows 10 Iot, Ubuntu and the official Raspbian OS.

- With RaspberryPi 4B it's better to use Raspbian OS due to the compatibility with the processor which is called ARM Cortex-A72 (ARM V8) which causes some problems with the other Operating Systems.
- Python 3.7 is installed and used as the programming language for the whole Software implemented in the Micro-Computer which sends and collects the data from different system components and controls the motion of the patrol robot.
- RPI.GPIO package is used to control Input/Output Pins of the RaspberryPi to control the motion of the patrol robot.
- Socket package is used to control the messaging between the RaspberryPi and the other System Components.
- Other Packages were used with Python 3.7 which is selected due to its stability and compatibility with most of the packages. -

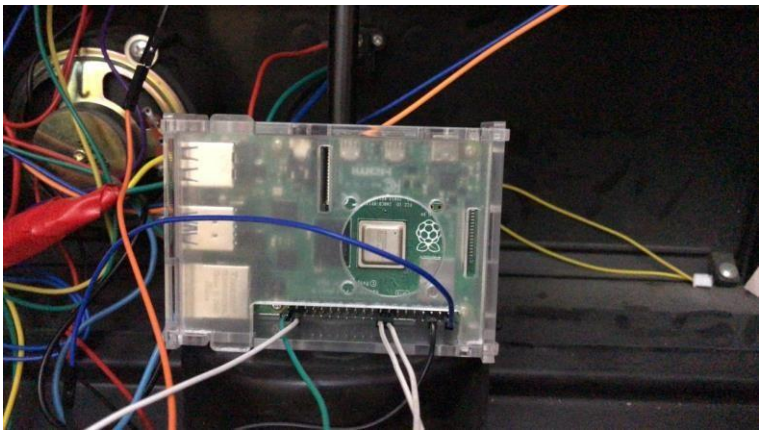


Figure [5-4]

5.3 Robot’s Hardware Structure

The robot’s hardware played a big challenge in assembling the parts and designing the suitable circuits for the communication between the software modules and the hardware parts represented in the DC motors.

The robot has three DC motors; the first two are responsible for the robot’s forward and backward movement, which is noted as the throttling movement. The third motor is responsible for the steering, it rotates both ways allowing for bidirectional steering with ease.

The robot is supplied with two 6-volt batteries connected in series, which results in total of 12-volt power supply, the motors a fed by the current resulted from the batteries, the two throttling motors are connected together in series along with the steering motors allowing all the motors to receive the 12 volts from the battery.

The challenge presented in controlling the feed of current to the batteries with the micro-computer, as the maximum value a micro-computer can handle in a circuit is 3 volts, and it cannot provide more that the same value which is 3-volts, so a mediator is required to fulfill this mission, so we used an H-bridge circuit.

So, we did connect the throttling motors together in series and treated them as a single motor so we would guarantee the syncing between them, and then we connected their feeder node to the H-Bridge, then we connected the steering motor with the second end of the H-bridge, then we used the PWM (Pulse Width Modulation).

Pulse-width modulation (PWM), or pulse-duration modulation (PDM), is a method of reducing the average power delivered by an electrical signal, by effectively chopping it up into discrete parts. The average value of voltage (and current) fed to the load is controlled by turning the switch between supply and load on and off at a fast rate. The longer the switch is on compared to the off periods, the higher the total power supplied to the load.

Along with maximum power point tracking (MPPT), it is one of the primary methods of reducing the output of solar panels to that which can be utilized by a

battery. PWM is particularly suited for running inertial loads such as motors, which are not as easily affected by this discrete switching, because their inertia causes them to react slowly. The PWM switching frequency must be high enough not to affect the load, which is to say that the resultant waveform perceived by the load must be as smooth as possible.

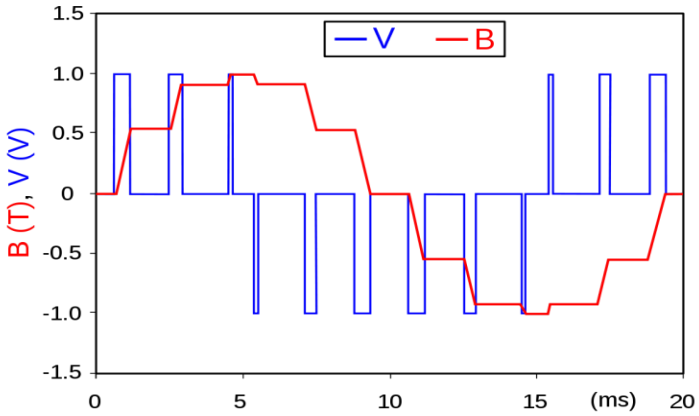


Figure [5-5]

Controlling the signals received from the micro-computer to the H-bridge which opens the switch allows the current to pass from the battery to the desired motor producing the required movement.

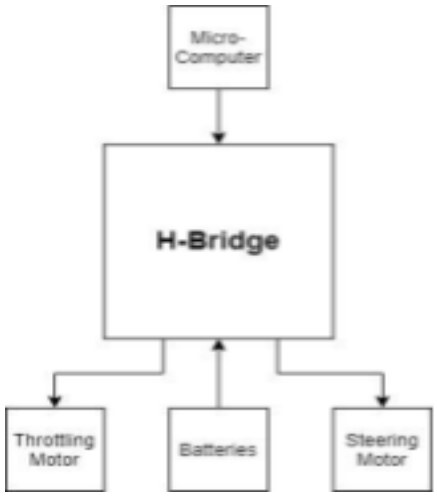


Figure [5-6]

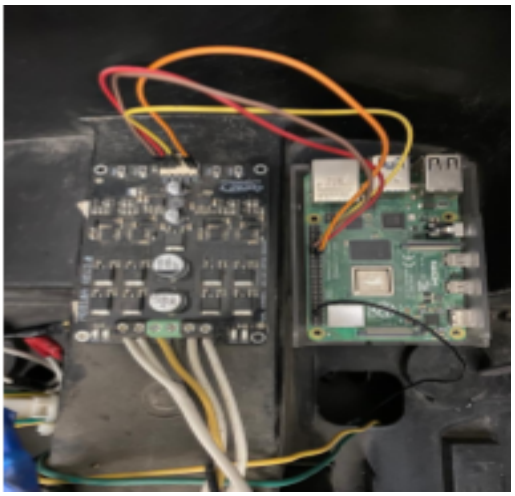


Figure [5-7]

And the outer body of the robot played a big challenge that we faced to make it suitable for the police force and here is the final look

Figure [5-8]

Chapter six
System Testing

System Testing

- Start the robot, by pressing on the power button to operate the motors

Figure [6-1]

- Start the Raspberry Pi by connecting the power bank attached to it.

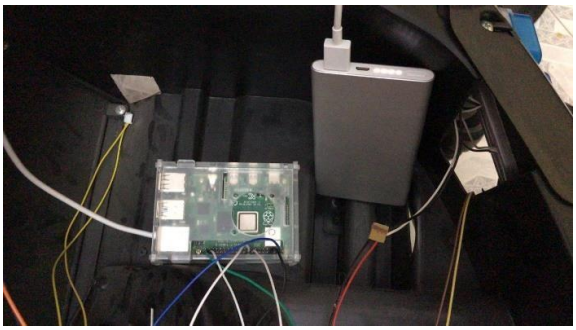


Figure [6-2]

- Open the smart delivery application which is installed on your device to start making your order and get your QR-code to receive your order.

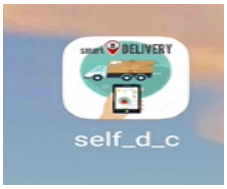


Figure [6-3]

- Open APP and select the route (Destination) that you wish the robot to navigate then export the route file.

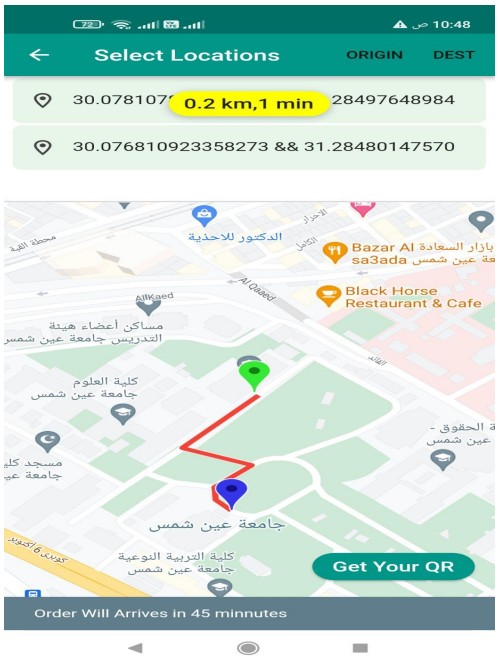


Figure [6-4]

- sending the frames from the raspberry pi camera to the system run the car modules and visual perception modules, the system should receive the frames and start to apply the visual perception and car modules on the frames.

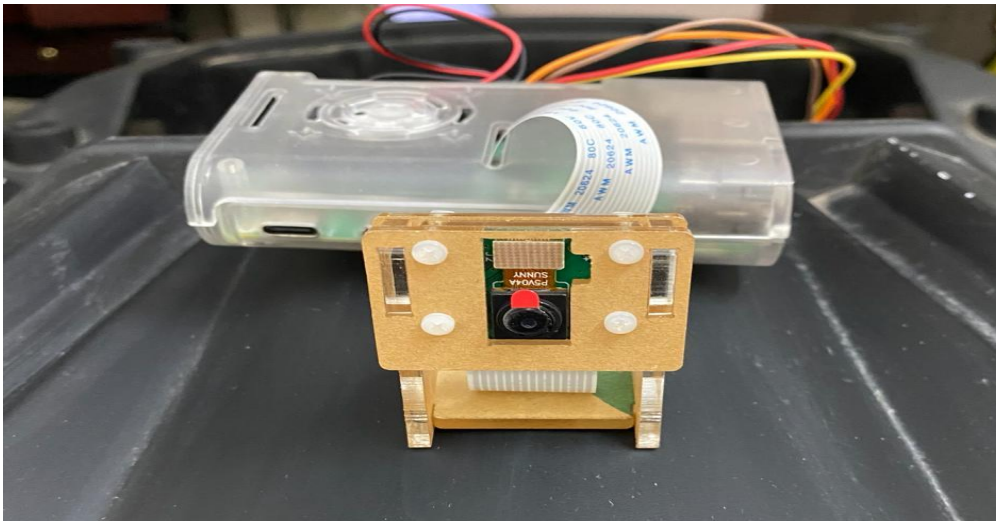


Figure [6-5]

- After finishing 1 frame the system will send a message and the output of the modules to the raspberry pi which this process is repeated for every frame, The robot should be able to receive it's coordinates from GPS sensor to start the localization map which will be sent to the raspberry pi.

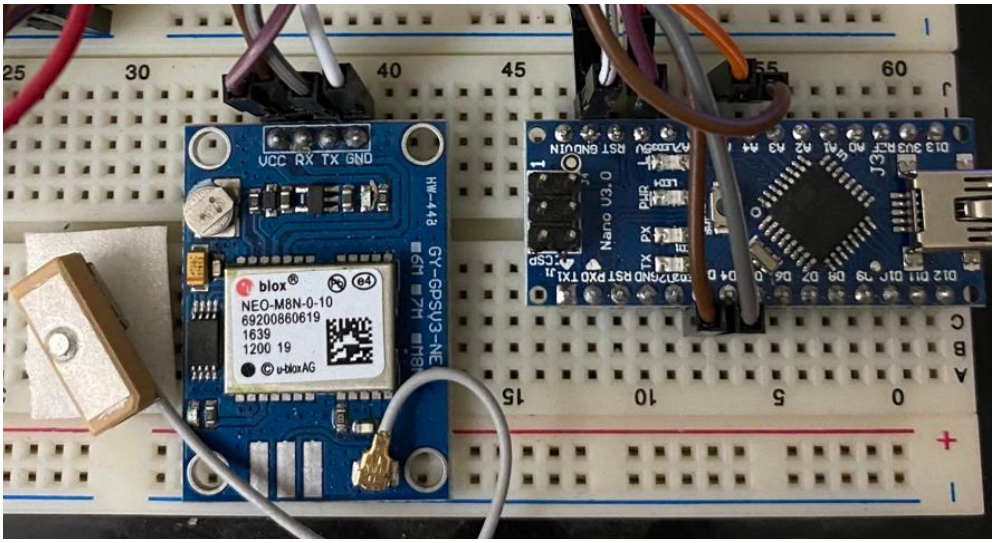


Figure [6-6]

- The raspberry pi now receives signals, output from sensors, mission map, ultrasonic, output from visual perception modules.

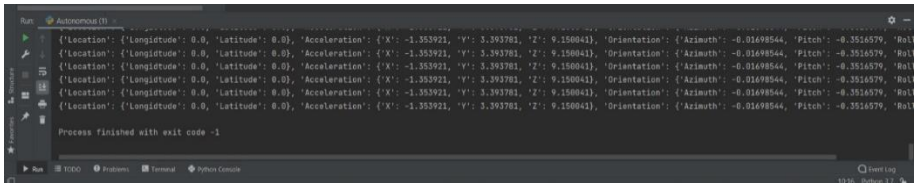


Figure [6-7]

- Raspberry pi will start to send visual perception output to user interface and coordinates, mission map, ultrasonic and localization output to the motion planner (PID).
- You will be able to observe the robot's vision of the area.

Chapter Seven
Conclusion and Future Word

Conclusion

After finishing the implementation and testing the project, we found that investing more effort and research in the robotics and artificial intelligence fields is a must as the variety of applications that can be implemented using this is huge, also to join and compete in the world race in the robotics and the artificial intelligence fields.

The self-driving delivery cars have proven its efficiency and the superiority that the technology can offer for us in order to make our life better and safer, as its abilities did far exceed the abilities of humans.

Implementing the visual perception modules and using complex artificial intelligence algorithms and techniques required studying a lot of papers and research whom most of them were foreign, and gaining the knowledge required for finishing the project mostly dependent on universal materials, so focusing more on the artificial intelligence field and affording more national effort is a must for us to compete and join the advanced nations.

Proceeding with the autonomous driving system, it has showed us the new direction the world's largest companies and universities are heading to which is the self-driving cars industry, which is the new trend in the world right now, studying a small part of it, which is the autonomous movement, allowed us to have a peak on this large technology allowing us to explore and conclude that the huge importance and impact of the self-driving cars covers both the human-benefits aspect, and the market aspect, and seeing that in the few upcoming years the world will see a huge transformation in this field.

Future Work

As for the success in the current phase of the project, the project can be taken into higher levels by improving the autonomous system capabilities and introducing higher forms and definitions for system localization and motion planning.

Taking the whole project idea, we can get the best opportunity to get the most out of the system, by applying the idea at compounds, modern cities, hospitals, clubs, etc, as it will give better coverage.

Figure [7-1]

For improving the autonomous system's visual perception capabilities, adding Lidar camera will help in improving the robot’s localization, use high specification computer to deploy the visual perception modules.

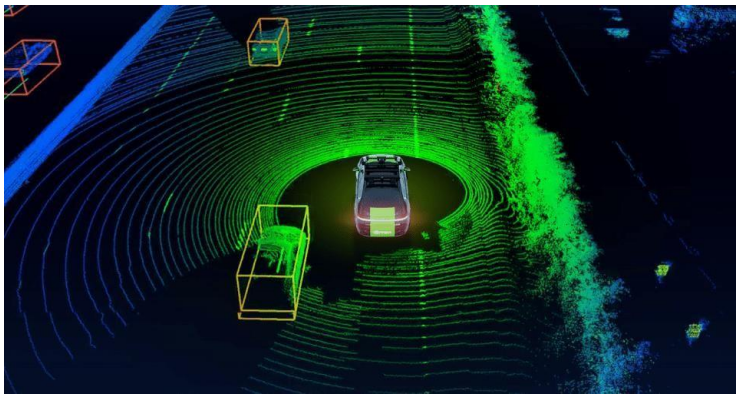


Figure [7-2]

Tools

Hardware:

1) Electric Car
2) Raspberry Pi 4B
3) Raspberry Pi Camera 8 Megapixel
4) H-Bridge
5) GPS, IMU, Accelerometer, UltraSonic
6) 2 Devices for Computation Software:
    1) Python Libraries:
        1) Numpy
        2) Pandas
        3) TensorFlow 2.5
        4) Pytorch
        5) Cuda
        6) Pillow
        7) Opencv
        8) H5py
        9) RPI.GPIO
        10) Socket
        11) Imagezq
    2) Flutter application
        1) flutter packages
        2) flask for local server
        3) firebase

References

[1] Visual Simultaneous Localization and Mapping,

[2] https://www.aboutamazon.com/news/transportation/meet-scout

[3] Society of Automotive Engineering official website.

[4] Warrandale, "Levels of Driving Automation", SAE Official website, December 2018.

[5] Einicke, "Smoothing, Filtering and Prediction: Estimating the Past, Present and Future (2nd ed.)", Amazon Prime Publishing, 2019.

[6] Julier, S.J.; Uhlmann, J.K., "Unscented filtering and nonlinear estimation" (PDF). *Proceedings of the IEEE*. 2004.

[7] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg "SSD: Single Shot MultiBox Detector V5", Cornell University, December 2016.

[8] Renu Khandelwal, "Single Shot Detector for Object Detection Using Multi-Box", Towards Data Science, November 2019.

[9] Abhilash Sachin Kulkarni, Jyothi S. Nayak, Aditi Desai, Jahnavi Singh, Shraddha Murali, " Deep Semantic Segmentation for Self-driving Cars", Egyptian Knowledge Bank, January 2021.

[10] Cheng, Yizong (August 1995). "Mean Shift, Mode Seeking, and Clustering". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. August 1995,

[11] Russell, Stuart J., "Artificial intelligence a modern approach". Norvig, Peter (4th ed.). Boston: Pearson, 2018.

[12] Jeremy Cohen, "Control Command in Self-Driving Cars", August 2018.

[13] SSD object detection: Single Shot MultiBox Detector for real-time processing paper.

[14] "GPS: Global Positioning System (or Navstar Global Positioning System)" Wide Area Augmentation System (WAAS) Performance Standard, Section B.3, Abbreviations and Acronyms, January 2012.

[15] Iosa, Marco; Picerno, Pietro; Paolucci, Stefano; Morone, Giovanni (2016). "Wearable inertial sensors for human movement analysis". *Expert Review of Medical Devices*. **13** (7): 641–659, May 2016.

[16] Tinder, Richard F. (2007). Relativistic Flight Mechanics and Space Travel: A Primer for Students, Engineers and Scientists. Morgan & Claypool Publishers. p. 33, June 2007.

[17] Danny Jost, "What is an Ultrasonic Sensor?", Fierce Electronics, October 2019.

[21]https://www.researchgate.net/publication/347383_The_Control_Structure_for_DC_Motor_based_on_the_Flatness_Control

[22]https://www.researchgate.net/publication/317225711_H_Bridge_DC_Motor_Driver_Design_and_Implementation_with_Using_dsPIC30f4011

[23] GPX: the GPS Exchange Format, Official Website, 2014.

[24] Lehmann, E. L.; Casella, George, "Theory of Point Estimation" (2nd ed.). New York: Springer, 1998.

[25] Sakarovitch, Jacques, "Elements of automata theory", 2009

[26] "Tiny Yolo", YOLO Official Website, April 2020.

[27] Qiao, Yu , "THE MNIST DATABASE of handwritten digits", 2007.

[28] A. Das, M. Wasif Ansari and R. Basak, "Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV," 2020 IEEE 17th India Council International Conference (INDICON), 2020, pp. 1-5, doi: 10.1109/INDICON49873.2020.9342585.

[29]https://www.researchgate.net/publication/323589806_Convolutional_Neural_Networks_Based_Fire_Detection_in_Surveillance_Videos

[30] Densely Connected Convolutional Networks (Gao Huang∗, Zhuang Liu∗, Laurens van der Maaten, Kilian Q. Weinberger)