DESIGN AND DEVELOPMENT OF CAMERA MODEL SYSTEM FOR OTORAYA AUTONOMOUS VEHICLE (OAV) BASED ON YOLO APPLICATION ON 1/10 TRACK

MAZEN OMAR MOHAMED MAHMOUD

Supervisor:
AHMAD HUMAIZI BIN ROZAINI

Faculty of Mechanical Engineering
2025

[This Page Has Been Left Blank]

DESIGN AND DEVELOPMENT OF CAMERA MODEL SYSTEM FOR OTORAYA AUTONOMOUS VEHICLE (OAV) BASED ON YOLO APPLICATION ON 1/10 TRACK

MAZEN OMAR MOHAMED MAHMOUD

UNIVERSITI TEKNOLOGI MALAYSIA

UTM(FKM)-1/02

School of Mechanical Engineering

Universiti Teknologi Malaysia

VALIDATION OF E-THESIS PREPARATION

Title of the thesis : DESIGN AND DEVELOPMENT OF CAMERA MODEL SYSTEM FOR OTORAYA AUTONOMOUS VEHICLE (OAV) BASED ON YOLO APPLICATION ON 1/10th TRACK

Degree: Bachelor of Mechanical Engineering with Honours

Faculty: Faculty of Mechanical Engineering

Year: 2025

I MAZEN OMAR MOHMAED MAHMOUD

(CAPITAL LETTER)

declare and verify that the copy of e-thesis submitted is in accordance to the Electronic Thesis and Dissertation's Manual, Faculty of Mechanical Engineering, UTM

(Signature of the student)

Permanent address:

22, Jalan Pulai Perdana
3/3, Taman Pulai Perdana
Johor Bahru

(Signature of supervisor as a witness)

Name of Supervisor: Dr. AHMAD HUMAIZI ROZAINI

Faculty: FKM

Note: This form must be submitted to SKM, UTM together with the CD.

[This Page Has Been Left Blank]

UNIVERSITI TEKNOLOGI MALAYSIA
DECLARATION OF PROJECT REPORT

Author's full name	:	MAZEN OMAR MOHAMED MAHMOUD			
Student's Matric No.	:	A20EM4015	Academic Session	:	202420251
Date of Birth	:	02/02/2003	UTM Email	:	mazen@graduate.utm.my
Project Report Title	:	DESIGN AND DEVELOPMENT OF CAMERA MODEL SYSTEM FOR ELECTRIC OTORAYA AUTONOMOUS VEHICLE (OAV) BASED ON YOLO APPLICATION ON 1/10th TRACK			

I declare that this project report is classified as:

☒	**OPEN ACCESS**	I agree that my project report to be published as a hard copy or made available through online open access.
☐	**RESTRICTED**	Contains restricted information as specified by the organization/institution where research was done. *(The library will block access for up to three (3) years)*
☐	**CONFIDENTIAL**	Contains confidential information as specified in the Official Secret Act 1972)

(If none of the options are selected, the first option will be chosen by default)

I acknowledged the intellectual property in the project report belongs to Universiti Teknologi Malaysia, and I agree to allow this to be placed in the library under the following terms :

1. This is the property of Universiti Teknologi Malaysia
2. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of only.
3. The Library of Universiti Teknologi Malaysia is allowed to make copies of this project report for academic exchange.

Signature :

Signature of Student:

Full Name: MAZEN OMAR MOHAMED MAHMOUD
Date : 19/2/2025

Approved by Supervisor(s)

Signature of Supervisor I:	Signature of Supervisor II
DR. AHMAD HUMAIZI BIN ROZAINI	Full Name of Supervisor II

NOTES : If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction

[This Page Has Been Left Blank]

“I hereby declare that I have read this report and in our opinion this report is sufficient in terms of scope and quality for the award of the degree of Bachelor of Mechanical Engineering with Honours”

Signature	:	
Name of Supervisor I	:	DR AHMAD HUMAIZI
Date	:	JANUARY 19, 2025
Signature	:	
Name of Supervisor II	:	SECOND SV
Date	:	JANUARY 19, 2025
Signature	:	
Name of Supervisor III	:	THIRD SV
Date	:	JANUARY 19, 2025

[This Page Has Been Left Blank]

DESIGN AND DEVELOPMENT OF CAMERA MODEL SYSTEM FOR OTORAYA AUTONOMOUS VEHICLE (OAV) BASED ON YOLO APPLICATION ON 1/10 TRACK

MAZEN OMAR MOHAMED MAHMOUD

A report submitted in partial fulfilment of the
requirements for the award of the degree of
Bachelor of Mechanical Engineering with
Honours

Faculty of Mechanical Engineering
Universiti Teknologi Malaysia

FEBRUARY 2025

[This Page Has Been Left Blank]

DECLARATION

I declare that this report entitled *“DESIGN AND DEVELOPMENT OF CAMERA MODEL SYSTEM FOR OTORAYA AUTONOMOUS VEHICLE (OAV) BASED ON YOLO APPLICATION ON 1/10 TRACK”* is the result of my own research except as cited in the references. The report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature	:	
Name	:	MAZEN OMAR MOHAMED MAHMOUD
Date	:	JANUARY 19, 2025

ACKNOWLEDGEMENT

With the blessing of Allah, the Supreme, the Very Compassionate. Thanks to Allah, the majestic Lord of the universe, if it wasn't for his blessing of strength, comprehension, and resilience, I wouldn't have made it through this last year report of Mechanical Engineering degree. Invaluably, I have truly learned to appreciate all the wonderful things that this path has been kind enough to offer me throughout this transition process.

In carrying out this writing, I have been fortunate to be around, and been able to socialize with, many interesting people of great caliber, including acclaimed scholars, focused educators, and skilled workmates. Cultural diversity has truly played a big role in my academic career as my perspectives and capabilities in this course have been greatly improved by their unique insights. I make a deep bow towards each of them and hold a dear place in my heart.

My gratitude is limitless, especially to my supervisor Dr. Ahmad Humaizi bin Rozaini, for his ceaseless motivation, insightful guidance, and consistent direction. Without him contributing ideas and working tirelessly this research could not have developed to where it is right now. In addition, I'm also extremely thankful to my Academic Advisor, Dr. Aminuddin Bin Saat, who has given me unwavering assistance and guidance in my academics in (Universiti Teknologi Malaysia) UTM.

I am highly grateful to UTM for the provision of space and materials which have enabled me to craft this unforgettable research. The academic syllabus and the learning fundamentals which undoubtedly have been instilled in me during these four years, it's been a moving factor in the production of this thesis.

On top of that, I am tremendously thankful to my guardian angel and my first teacher, my father Mr. Omar Mohamed, who supported my ideas and gave valuable advice without any changes in order to achieve my life goals. His coaching was instrumental to my strong performance. But most all of all, I would like to thank my mother who is the person who has always been there to give a safe and warm environment to me.

To finish up, let me take the opportunity to appreciate my dear friends who would always be my brothers. Arslan Babur and Ahmed MohamedAli. Individuality the family between us, we have met both the good and the difficult moments with constancy, becoming stronger together every time. Their friends and mates also play a vital role by providing a fortress and stimulating them in every way possible.

On conclusion, I could only thank with all my heart everyone that been so supportive in the writing of the thesis and in the support that he gives me to come here. I pray Allah increases them greatly.

ABSTRACT

The development uses artificial intelligence perception to construct an autonomous vehicle on a 1/10th scale that obtains Level 5 autonomy through system integration of navigation and perception capabilities. Image perception operations benefit from the accurate detection system created through the combination of OpenCV and YOLOv8. The management of motor operations functions through two systems. The Raspberry Pi controls low-level procedures but the Arduino system handles vision-based operations. The camera system records high-definition environmental information that enables automatic detection of track boundaries along with features as well as obstacle recognition with a response time under 5 seconds on average. The servo motor uses the DC motor to drive the vehicle up to 10 km/h speed with precise steering capabilities. Real-time mapping capabilities of the vehicle system when paired with ultrasonic sensors enable the system to perform proximity detection functions for secure operation. The performance testing system needed proper development of a track to simulate demanding operational scenarios including curved paths and safety obstacles in limited operational boundaries. The detection system failed to achieve accurate results when operating at low-light or during unstable routes because of integration challenges and environmental factors which limited its precision. The reliability standards of sensor fusion approaches require adaptive learning improvement to meet performance standards along with faster image processing rates that uphold those standards. The autonomous navigation project functions as a robotic control technology development while delivering practical benefits toward education for students in robotics and artificial intelligence throughout smart transportation systems. Modern advanced technologies enabled engineers to develop an autonomous system which supports the creation of self-governed urban transportation networks.

ABSTRAK

Pembangunan ini menggunakan persepsi kecerdasan buatan untuk membina kenderaan autonomi berskala 1/10 yang mencapai autonomi Tahap 5 melalui integrasi sistem navigasi dan persepsi. Operasi pemprosesan imej mendapat manfaat daripada sistem pengesanan tepat yang dihasilkan melalui gabungan OpenCV dan YOLOv8. Kawalan operasi motor berfungsi melalui dua sistem, di mana Raspberry Pi mengendalikan prosedur peringkat rendah, manakala Arduino mengurus operasi berasaskan penglihatan. Sistem kamera merakam maklumat persekitaran berdefinisi tinggi, membolehkan pengesanan automatik sempadan trek, ciri-ciri laluan, dan halangan, dengan masa tindak balas purata kurang daripada 5 saat. Servo motor mengawal motor DC untuk menggerakkan kenderaan dengan kelajuan sehingga 10 km/j, memberikan keupayaan stereng yang tepat. Keupayaan pemetaan masa nyata yang digabungkan dengan sensor ultrasonik membolehkan sistem melaksanakan fungsi pengesanan jarak bagi memastikan operasi yang selamat. Sistem ujian prestasi memerlukan pembangunan trek khas bagi mensimulasikan senario operasi mencabar, termasuk laluan berselekoh dan halangan keselamatan dalam kawasan operasi yang terhad. Walau bagaimanapun, sistem pengesanan gagal mencapai ketepatan yang tinggi apabila beroperasi dalam keadaan cahaya rendah atau laluan yang tidak stabil disebabkan cabaran dalam integrasi sistem dan faktor persekitaran yang menjejaskan ketepatan pengesanan. Kaedah gabungan sensor memerlukan peningkatan pembelajaran adaptif bagi memenuhi piawaian prestasi, bersama dengan kadar pemprosesan imej yang lebih pantas untuk mengekalkan tahap kecekapan yang diperlukan. Projek navigasi autonomi ini bukan sahaja berfungsi sebagai pembangunan teknologi kawalan robotik, malah menyediakan manfaat praktikal dalam pendidikan bagi pelajar dalam bidang robotik dan kecerdasan buatan melalui sistem pengangkutan pintar. Teknologi moden yang maju membolehkan jurutera membangunkan sistem autonomi yang menyokong pembentukan rangkaian pengangkutan bandar yang dikendalikan sendiri, membawa perubahan ke arah mobiliti pintar dan mampan.

TABLE OF CONTENTS

LIST OF TABLES

TABLE NO.	TITLE	PAGE

LIST OF FIGURES

LIST OF ABBREVIATIONS

PC	-	Personal Computer
UTM	-	Universiti Teknologi Malaysia
WWW	-	World Wide Web
NMS	-	non-maximum suppression
IoT	-	internet of things

LIST OF SYMBOLS

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background of Study

In the 21st century context, the merging of smart track infrastructure with autonomous vehicles is a symbol of modernization and evolution. While we embark on the venture in which we develop the 1/10 scaled model track for Octorara Autonomous Vehicles (OAVs), it is critical to examine the value of these innovations and describe the plan for our project execution.

Importance of Advanced Track and Autonomous Vehicles: In the cities, that advanced lead equals the pinnacle of urban development when all the benchmark technology and sustainable lifestyle combine in order to upward the quality of life for the inhabitants. The introduction of autonomous vehicles in the ecosystem is a sort of change in basic assumptions in transportation, which will provide safety, efficiency, and environment sustainability for individual users, the public and the planet. As a team, our endeavor is twofold: to develop both the simulated track model and the monitoring system for electric OAVs. This dual approach is aimed at creating an environment that mirrors the intricacies of real-world urban landscapes while leveraging state-of-the-art technology to propel us into the future.

Developing the Advanced Track and Autonomous Vehicles: Apart from that, with the aim of building a simulated smart track as previously mentioned, our approach includes a detailed study about successful smart cities that are already existing. Through recognizing and regarding the applicable lessons from fellow urban leaders like Singapore, Barcelona, and Amsterdam, we will establish a model which sets a new target to be obtained.

Key features of our simulated track include: • Instalment of the CCTV cameras and sensors at the traffic lights to monitor vehicle speed and detect accidents in real time. • Integral implementation of smart traffic management system to manage traffic, improve movement, and decrease the jams. • Use of auto crash alert mechanism that notifies the police department and other emergency services if accidents and violations occur very quickly. • On the other hand, use renewable energy resources, efficient general refuse system, and green areas to educate the society and make it more conscious of the environment. Thus, the replication of the actual urban system in our simulated track will not only mimic real track patterns but also allow for trial and error of urban issues through the usage of innovative solutions that make the track smarter.

Utilization of YOLO for OAVs: Finally, I want to emphasize that my college journey has not only opened my eyes to new dimensions of society and strengthened my perspectives, but it has given me the chance to establish connections with individuals of diverse backgrounds. We will deploy YOLO (You Only Look Once) algorithm on the image object detection and reaction, and that will be the cornerstone of our autonomous vehicle technology. YOLO represents a new perspective on the real object recognition, based on deep learning algorithms and extensive database with an aim to analyze environment and process of data instantly. In essence, YOLO enables OAVs to: • Sensors and responsive algorithms in place to determine and react to sudden environmental factors during the process, thus adequate decision-making can happen real time. • Enable people without driving license and disables who are not able to drive to access public transit. Provide mobility to others who have difficulties walking or carrying heavy bags. • Safety of the citizens should not be compromised, hence the ways to prevent accidents and security breaches prior to their occurrence should be considered. Eventually, the track area will be secure for comfortable living. Given the fact that YOLO is making the markets improve, we are planning for the future where autonomous vehicles will become a part of the transportation system fabric and will progressively transform safety and driving standards.

1.2 Problem Statement

The implementation of our project, which involves the creation and realization of a scaled 1/10 model system of Otoraya Autonomous Vehicles (OAVs), has presented some major challenges that need careful thought and analysis.

Initially, the procurement of critical materials, including motherboards, Jetson, Raspberry Pi, processors, cameras, and other equipment needed for vehicle testing is the greatest difficulty. The aforementioned components make up the foundation of our monitoring system and vehicle simulation and therefore their availability and affordability are the key to successfully implementing our project.

However, putting together a wide range of separate elements into a unified scheme is a multi-faced problem. The project's success relies not only on the performance of the monitoring system but also on the smooth working of mechanical and design features. This integration involves a combination of both technical and artistic considerations as well as an integrated approach, which makes the development process even more complex. On top of this, the construction of a virtual track makes the project even more elaborate. Creating realistic urban environments requires the reproduction of fine details like intelligent traffic signals, security systems, and interactive infrastructure. The accomplishment of this level of fidelity using intricate research, planning and implementation is increasing the project's technical tasks.

Another considerable obstacle is the educational aspect of the project. The objective of the production is to make the assembly easy to assemble and understandable by the audience of teenagers. Simplifying assembly and keeping functionality and educational value at the same time creates a challenge that demands creativity and consideration of a user interface. In other words, our project will need to manage obstacles from procuring crucial components to bringing together different technologies and making the assembly process available to everyone. Dealing with these challenges would require a collaborative and multidisciplinary approach involving experts in engineering, design, education, and project management to complete the task fully while performing all the functions correctly.

1.3 Research Objectives

Objectives

a. To design and develop 1/10 track model system for OTORAYA autonomous (OAV) based on YOLO application.

b. To develop 1/10 size CAMERA MODEL OTORAYA autonomous (OAV) based on yolo application.

1.3.1 Scope

Research scope

a. The prototype CAMERA MODEL system for Electric OTORAYA Autonomous Vehicle (EAV) for 1:10 scale.

b. The track model for Electric OTORAYA Autonomous Vehicle (EAV)) for indoor usage only.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In the fast-changing urban environment of today, where advanced track infrastructures gradually mix with autonomous vehicle technology, the way we lead our lives and move around cities is also experiencing transformation. ”Intelligent cities” leverage cutting-edge technologies like IoT and AI to build smarter and more efficient urban centers. Self-driving vehicles, which are packed with sensors and AI technology, are designed to be safer and more comfortable for people to travel.

This transition, nonetheless, is accompanied by difficulties. Regulatory barriers, infrastructural needs, and social acceptance act as major hindrances. Incorporating self-driven cars into the structure of smart cities necessitates a comprehensive approach including urban planning, policymaking, community engagement, and so on.

Our tasks are to design a 1/10 scaled track model system for Otoraya Autonomous Vehicles (OAVs) simulated track that replicates the real-world urban dynamics but use technology for future driving systems. Through thorough research and creative solutions, we try to harmonize theory and practice, which results in a better, safer, and greener urban environment.

2.1.1 Autonomous Vehicles

The goal of our project is to become the forerunners in autonomous driving and finally to develop an autonomous vehicle that can completely do away with any human intervention and reach the level 5 autonomy. This is an embodiment of the highest

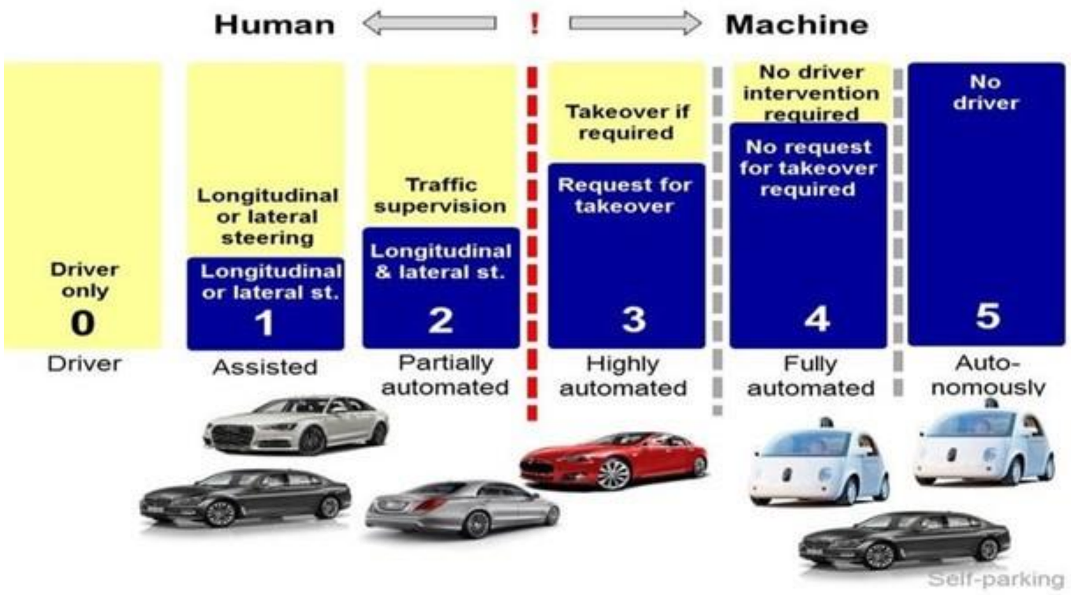


Figure 2.1 Levels of the autonomy (*Levels of Autonomous Driving According to SAE J3016. | Download Scientific Diagram*, n.d.)

degree of autonomous vehicle intelligence, which is the ability to perceive, interpret and communicate with the environment without any direction by the driver.

For us to achieve this ambitious target, there should be an integration of a variety of technologies and methodologies of computer programming. At Level 5 the vehicle is required to be fitted with a state-of-the-art suite of sensors comprising of vision sensors, Lidar, radar, and ultrasonic sensors, to have a 360 degrees perception. This set of sensors will give a real time information about the car environment that will in turn help it in managed decisions and safe navigation.(Zamindar, n.d.)

The central objective of the development of an autonomous vehicle is in the successful accomplishment of the detection and reaction features which should be faster and more efficient. This therefore requires the implementation of a few particularly strong processors and high-performance computing units to process large volume of sensor data in a short period of time. Also, brilliant algorithms and machine learning techniques will be used to decode sensors' input and to predict future potential hazards or obstacles in front of the vehicle.(Washington & Diehl, 2018)

The heart of the autonomy system of autonomous vehicles is the integration with the sophisticated control system. The system involves meaningful and flawless cooperation of data from different sensors and their processing for real-time actions. Moreover, high levels of reliability and robustness are needed, including the capability to tackle unforeseen cases and thus guarantee the safety of all subjects, including passengers and pedestrians.(Bartneck et al., 2021)

In summation, the accomplishment of Level 5 autonomy for Otoraya Autonomous Vehicle may be classified as a formidable challenge that necessitates the integration of cutting-edge technologies, their advanced programming, and robust control systems. Through these tools and the technique, we will develop an autonomous vehicle with superior intelligence and outstanding performance in urban transportation, compared to others which will be of higher safety, efficiency, and reliability.

2.1.2 Autonomous Vehicles in Industry

In the struggle of Level 5 autonomy, some well-known automotive companies have done the most in allocating financial resources as well as progressing autonomous vehicle. We consider what approach and strategies major rivals, including Tesla, Volkswagen, Baidu, and Daimler, have taken to advance technology and sales figures are thoroughly gone over.

Tesla: Tesla has always been associated with self-driving innovations thanks to the young and dynamic CEO, Elon Musk. The actual determinants of Tesla's autonomy lie in the Autopilot and Full Self-Driving (FSD) operations. These systems consist of a fusion of the camera, ultrasonic sensors and the advanced network of neural networks in order to facilitate the driving process, as well as make it become completely autonomous. All vehicles of Tesla work as a massive data gathering device. All the outputs from vehicles are collected and fed back into Tesla's AI algorithms which are used for training and update. Tesla's software updates via over-the-air can easily be rolled out and are considered as the major factor that provide it with a competitive advantage in the market of self-driving cars. From the sales point of view Tesla has obtained tremendous success. Its EVs and innovative autonomy have made the customers favor this brand and provoke the market penetration.

Volkswagen: Audi, which represents Volkswagen are the telling case for level 5. Audi is the first automaker since the alliance of the VW Group, also invest into level two. Automotive industry has been involved in the field of autonomous driving, which requires greater sensors, radar, LiDAR, and cameras to obtain a full purpose interpretation. The modular electric platform of Volkswagen (MEB) is the basis for autonomous driving systems that provides easy input and output linkage of hardware and software throughout its entire vehicles product line. Agency of collaboration between automobile company, such as Argo AI its investments in AI research, supports the targeted development economic growth and adoption of autonomous mobility. While Volkswagen's sales performance in the electric and autonomous vehicle segments is growing, it faces stiff competition from established players and new entrants alike.(Bartneck et al., 2021)

Baidu: Despite Baidu's popularity in China as a search engine giant, it as well has assumed a leading position in the self-driving field with Apollo. Apollo is built upon an agglomeration of the latest technological innovations that involve different technologies like sensor fusion, high-definition mapping, and AI-based decision-making algorithms, which are essential in attainment of autonomous driving (Level 5 autonomy). Baidu's wide-ranged understandings of AI technologies as well as strategic partnerships ensuring with a great number of the world-leading automotive and high-tech companies help Baidu to take the top position among rivals in this self-driving race. In reference to the Baidu's market presence, it is worth admitting that the Chinese market is highly dominated by the company as it has managed to develop and integrate its autonomous driving technologies in the market where both consumers as well as the regulators have shown keen interests at them.

Daimler: Daimler, parent of Mercedes-Benz, is leading among manufacturers of autonomous driving vehicles, paying special attention to safety, operational reliability, and user-friendly interface. Daimler's ADAS and autonomous drive initiatives provide environment recognition and decision-making abilities with help of combination of LiDAR, radar and cameras which provides high preciseness perception. The company's total solution for autonomously moving people involves V2X communication, high-definition mapping along AI-based predictive analytics. The engineering expertise of Daimler and the brand which stands for premium driving

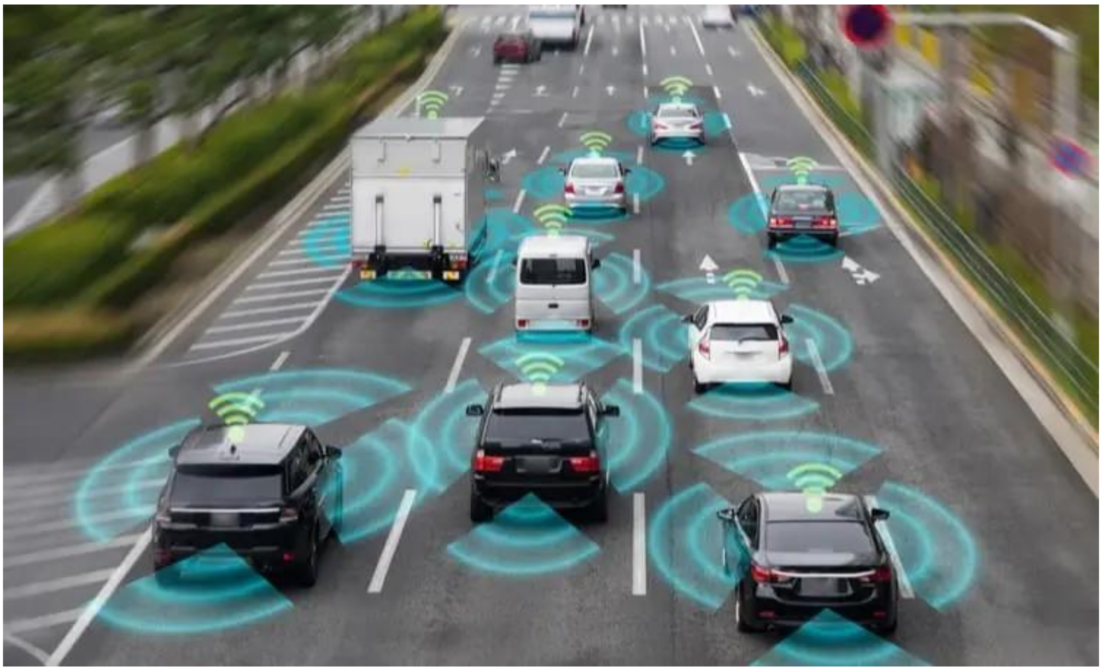


Figure 2.2 Autonomous Vehicle (*The Use of RADAR Technology in Autonomous Vehicles | System Analysis Blog | Cadence*, n.d.)

experience put it in the top positions within the automotive industry to promote the progress of autonomous driving technology. On the one side, Daimler enjoys the growing sales of the electric as well as autonomous vehicle products. But on the other, it is challenged by the powerful competitors coming from both the automobile sector and the tech field.

Comparative Analysis: All the competitors come with their own abilities and strategies to move forward and win in the competition to Upper Level 5 autonomy. Tesla's dogged pursuit of software generation and big data driven development is the main reason why it stands tall amongst the existing participants in the drive- lees technology that has greatly increased its sales and market leadership position. The fact that Volkswagen operates worldwide, has inter-group cooperation's, and has investments in the electrification and independent driving technology gives it an important position among many great achievements. Baidu's powerful research center and wide range of market control in China provides them with a great advantage to stand up against other around the world. Safety, engineering capability and premium driving experience are core values of Daimler, its market position is unshakeable within the luxury autonomous vehicle segment and its customer base is loyal and the reputation for innovation is worthy for mention. Proud of the extremely breaking-in the race for Level 5 autonomy, these corporations are the pioneers in tech sector, bringing some revolution to the transport industry.(Parekh et al., 2022)

Potential of Autonomous Driving Technology: Autonomous driving vehicles can give a vast potential for the transport industry which extends to the development of the cities and even to the whole humanity. At Level 5 Autonomy, cities can anticipate a 2.0 society; eliminating human errors and redeeming our transportation systems with next generation of buses and taxis that offer unmatched levels of security, effectiveness, and accessibility. Autonomous cars can alleviate traffic jams, improve air quality, and also boost traveling possibilities for the weak and the elderly. Besides, distributed driving automation is not only an innovation of business models such as healthcare and insurance, but also has chances in logistics, life delivery and last-mile transportation. Technology is not a still thing, so it continues to develop and make a way for itself. That is why the autonomous driving would get an incredible transformation as a tool of our cities and a way to make a new experience in traffic.

2.1.3 Detection and Autonomous Technology

In the domain of self-driving technology, companies like Tesla, Volkswagen, Baidu and Daimler are on the mainstream and their products usually work on the principle of sensing and operating, which are made of advanced hardware and software systems.

The statement that the leading car manufacturer Tesla embraced Autonomous Driving in practice is depicted by introducing Autopilot and Full Self-Driving. These systems, in turn, are supported by a wide range of technically complex hardware such as cameras, radar sensors, and ultrasound sensors, placed precisely throughout the vehicle to create an all-around view of its surroundings. Lesson learned by Telsa relies on complex processing of data inputs, drawn from such sensors and being transmitted to their deep learning systems based on neural networks. These neural network structures, together with the unique software stack Sens eye developed, ensure that real-time object detection, identification, and decision-making are carried out. The program code that Tesla uses for the autopilot is delicately crafted from a number of

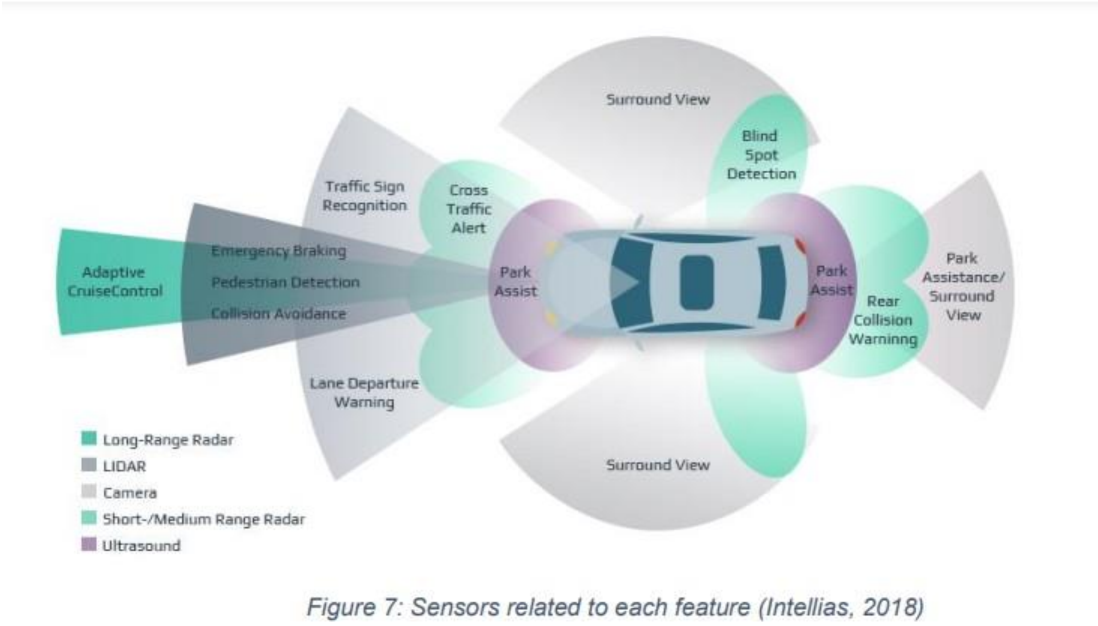


Figure 2.3 Sensors used for each feature (Buganza Co-Supervisors et al., n.d.)

programming languages including Python and C++, while TensorFlow is the force's backbone for the machine learning functions it performs.

Revealing Volkswagen's autonomous driving capabilities that are guaranteed with advanced hardware and software solutions, which makes this work of Audi is only a matter of time. By virtue of LiDAR, camera and radar sensors that are supposed to be integrated in the cars that are made by Volkswagen, these automobiles have the potential to accurately map and perceive the environment around them. The whole point of this software suite lies in the unification of data gathered by sensors, which are mixed together by sophisticated software algorithms, allowing to determine the exact environmental conditions and track objects. The deployment strategy of software development under VW relies on a mixed approach. This is typical as the programming languages like C++ and python are developed just as the specialized ones from the specific industry which are here for autonomous driving demand. As an autonomous car driving alliance between Baidu's Apollo and its state-of-the-art technological platform and the strategic partnerships with a higher aim to make driving autonomous mobility better, it is a representation of the combination of such technology and the partnerships. LiDAR systems, cameras, GPS and inertial measurement devices are the essential items of the equipment. These management systems are used to determine

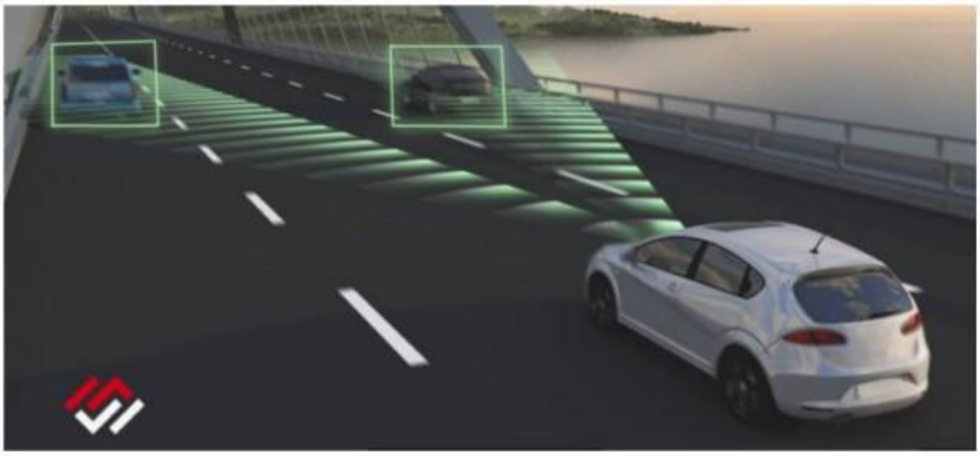


Figure 2.4 Car Navigation System (*How Does a Self-Driving Car See? | NVIDIA Blog*, n.d.)

changes in environment and device positioning and navigation. The ability of sensors to work with ML algorithms that can process a huge volume of the sensor data in RT stems from the ML algorithms which are designed to read the data and process it in the required times. The stable programming base of which Ba's software ecosystem is built on, which are mainly languages like Python and C ++, incorporates deep learning frameworks like Tensor Flow and Py Torch with the responsibilities shared among multiple modules which include planning and decision making.

Daimler, where Mercedes-Benz emanates from, its technical and marketing know-how as an affiliate partner, unveils the new autonomous driving technology positioning it at the forefront of the road safety, reliability, and user experience. The Vehicles with hardware actions such as fixed cameras, radar sensors, ultrasonic sensors, Vehicle to every (V2X) communication modules detect their environment fully and also get full-body diagnosis. Daimler places sensor fusion strategies (the method where the data coming in from different sensors is pooled for more reliable object drops and navigating the vehicle during turns) on its software architecture so that chances of mistakes are minimized. Machine learning algorithms do not bypass as they are the basic building. Machine learning algorithms play a pivotal role in adaptive driving decisions, leveraging a combination of C++, Python, and proprietary algorithms to enable sophisticated autonomous driving capabilities tailored to Daimler's vehicle platforms and applications. (Kumar & Mehar, 2023)

Primarily, among the companies pursuing the autonomous driving technology, the integration of the whole range of hardware and software solutions that have been pre-designed, is a very detail-oriented and tedious task as there are quite many elements, that are unveiled only at this stage of the development of a technology that is new and somewhat unprecedented, that seek to tackle the specific challenges that autonomous transportation faces. Autonomous vehicles are likely to become the new standard on roads just in few years. The reason is that the cutting-edge technology, working collectively, and implementation of the new systems take the control of the road rather following the old standards.(Martin, 2019)

2.1.4 Ethics related to Autonomous cars

Autonomous cars, praised as one of the future transport innovations that may change everything on our roads, have the potential to significantly improve road safety by easily getting rid of the human errors which are one of the main factors of accidents globally. While the unavoidable fact that some of the crashes will occur in turn provides ethical dilemmas that never should be ignored about what these vehicles should do when faced with an unavoidable crash. The ethical implications from which the crash algorithms have spurred philosophical inquiries raised questions like which life should be a priority, the extent to which different interests should be balanced, and the ideas of decision-making in an autonomous vehicle.(Othman, 2021)

One prominent ethical framework that has emerged in addressing this challenge is Contractarianism. contractarian approaches aim to derive crash algorithms that are acceptable to all rational agents, taking into account individual self-interest while seeking to reconcile conflicting preferences into a common solution. In the pursuit of an optimal algorithm, the principle of harm minimization often emerges as a central tenet. This principle seeks to minimize overall harm in a crash scenario, reflecting a utilitarian perspective that prioritizes the greatest good for the greatest number.

Although, issues, related to contractarian arguments also pinpoint the shortcoming of their application to crash algorithms. One crucial criticism faces the connotation of the situation: the crash relates, firstly, to practitioners and, secondly, to

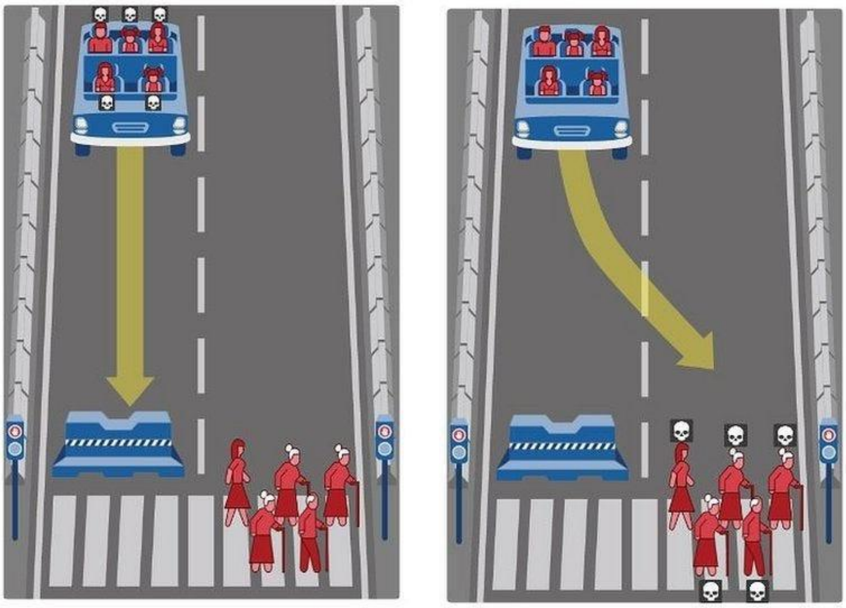


Figure 2.5 Trolley problem (*Out of Two Million People, Most Prefer That a Self-Driving Car Kill the Elderly*, n.d.)

those who are not part of it. Scholars speak of the trolley problem as an illustration of distinctions drawn between individuals that fall into the two categories: people who have positive obligations to help the one in need and others having simply a negative right not to be interfered with. A lack of clarification here could result in scenarios where the developed algorithms have insufficient understanding of the interests and the issues of that multiple stakeholders, resulting in substantial ethically inadequate conclusions.(Hussain & Zeadally, 2019)

Moreover, come to the translation of the principle of ethics to it in real life situation that is concrete into the algorithms which are computer software used in autonomous vehicles. The questions and debates raised by philosophers in their philosophical discussions, the ‘trolley problem ‘included, often fail to mimic the ambiguities and uncertainties of actual road crashes. On the other side of the coin, many ethical theories offer useful ideas, but their approaches to the application in autonomous driving technology should involve the resolution of contextual factors such as legal and social norms as well as technical filter.(Hübner & White, 2018)

Aside from technological advances and military ethics, the concept of liberal legitimacy likewise adds to the moral puzzle of crash algorithms. In any generic

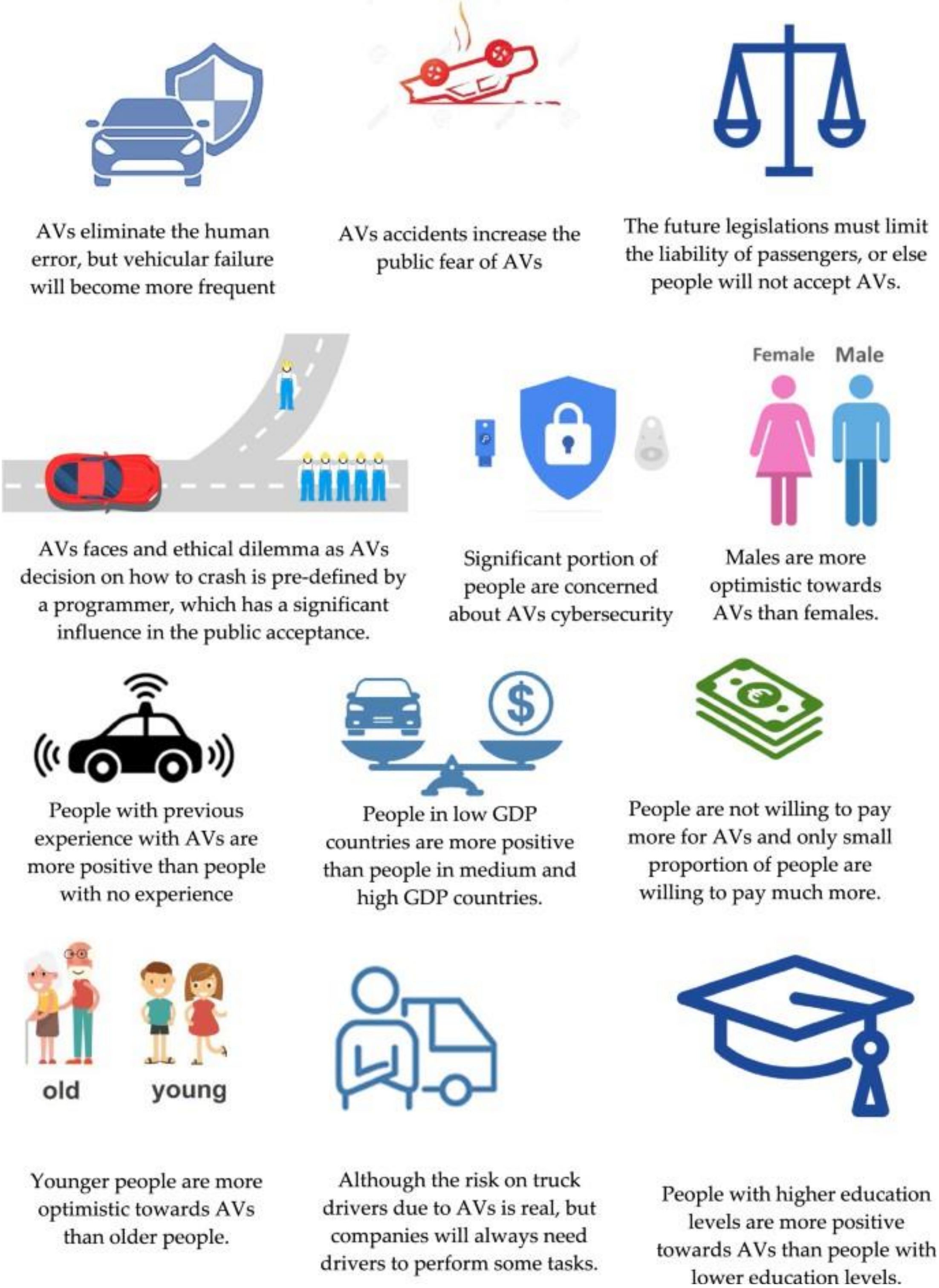


Figure 2.6 Problem faces Autonomous cars(Othman, 2021)

Point of View

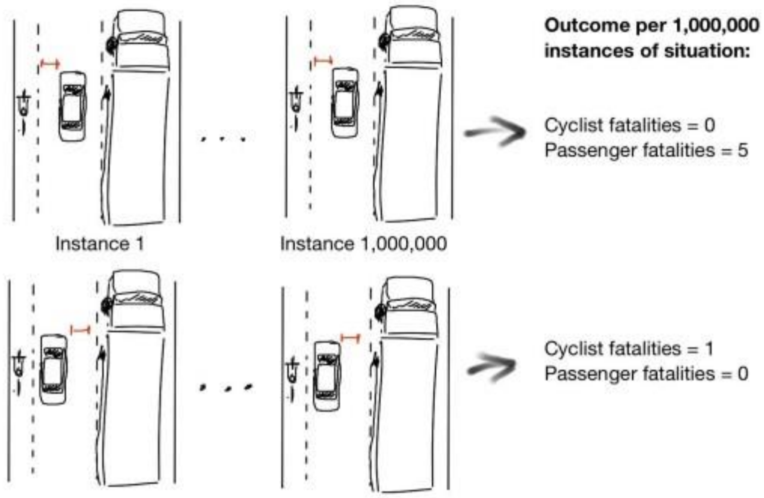


Fig. 1. ***Statistical trolley dilemma: a large truck appears in the lane next to a car with five passengers. The car must choose whether to stay put (top) or move slightly left (bottom). Aggregated over thousands of cars and millions of miles, small shifts in positioning can shift the relative risks incurred by different categories of road users (e.g., cyclists versus passengers).***

Figure 2.7 Trolley dilemma (Bonnefon et al., 2019)

domain, the rule that is mandated must be answerable to the principles of liberal neutrality. Therefore, it must be accepted by all the members of the society. On the other hand, it will be a difficult task of crafting a universal algorithm that satisfies different ethical points of views, which in itself, already illustrates the inherent dilemmas faced by a multidimensional society.

Though this will be not an easy task, a morally accurate autonomous driving, is ultimately the only right direction to follow. Ethical consultation is crucial as the autonomy technology gets more advanced, and the stakeholders must engage in continuous debates and ethical evaluations that minimum threats are taken. They should aim to come up with need-based algorithms that do not affect other groups in terms of fairness and the general societal advancement. Sculpting the future through the integration of moral principles into the design and ope-rationalization of self-driving vehicles involves striving for a time in which refinement and justice both shape transportation systems to make them ethically better.(Evans et al., 2020)

Developing crash algorithms for autonomous vehicles goes beyond the field of philosophy, law, engineering, and psychology. It requires a multidisciplinary approach that coordinates these different areas of expertise. There is a technical challenge prior to the implementation of ethics in software algorithms because the vehicles must be able to navigate ethical problems in a real time while also remaining in the legal and regulatory framework. Psychologists deal with the investigation of human subjective views of autonomous driving choice and moral decision making, exploring the process that can guide the design of algorithms that conform both to the basic social norms and people's expectations.(Hussain & Zeadally, 2019)

Besides this, people's participation, especially the representatives and experts on regulation of this technology, must be regarded as the final determining elements of the ethical development of independent vehicle technology. Creating forums for governmental and non-governmental organizations, as well as residents to participate in open dialogue is very important because these dialogues can yield useful insights on public perception of crash algorithms and ethical positions. Transparency and accountability mechanisms can be placed within an autonomous vehicle system to assist in inspection of decision-making processes and compare them with societal values.(Hübner & White, 2018)

In the end, the ethical dilemmas that cover the issue of crash algorithms for automatic drives are very complex and multi-sided, implying deep discussion about various conflicting values. Whilst contractarian perspectives may provide a sound theoretical bulletproof vest for these challenges, practical wisdom approach from a wide range of academic disciplines is needed to confirm and reinforce that autonomous vehicles uphold ethical standards as they make life safer on the road. In this way, we can effectively steer through the challenges tactfully and inclusively thus unleashing the wonderful potentials inherent in autonomous vehicle technologies in building safer and better transportation systems for everyone.

Figure 2.8 YOLOv8 (*Yolo V8: A Deep Dive Into Its Advanced Functions and New Features | by Mujtaba Raza | Medium*, n.d.)

2.2 Develop detection system Electric Otoraya Autonomous Vehicle (OAV) using YOLO

Towards the resultant of the gate detection system for the Otoraya Autonomous Vehicle (OAV), we have selected YOLOv8 (You Only Look Once version 8) as our definitive framework. In terms of its accuracy and efficiency, YOLOv8 is definitely one of the best options when prompted to choose a tool for capturing and processing images in real time.

Using YOLOv8 was in our focus due to several advantages, to be precise. First thing first, its advanced architecture gives the car the ability to identify objects with high precision. This is what makes the car robust in tracking objects in the car's vicinity. Furthermore, YOLOv8 can run with custom datasets, and we can implement our specific datasets during training and deployment, which means our detection solution can be easily tailored appropriately to our scenes. This feature is essential for the system to be able to conform in various ways to the sharp differences of genuine occurrences.

The YOLO collection of databases harbors a large number of images that are fully labeled and tagged individually. They denote the items or objects having the attribute like their categories present in the specific locations of an image. By training YOLOv8 on a wide range of images, it can correct its prediction in the test therefore, it improves its practice in identifying and classifying the objects that are unseen in the related images.(Arkadiusz & Biel, n.d.)

It's time to streamline the YOLOv8 and broaden the device's performance. Therefore, we will use OpenCV (Open-Source Computer Vision Library). OpenCV is a powerful library for image and video processing piping, like image manipulation, feature detection, and object tracking, are the list of tasks which it performs.

In a nutshell, the combination of OpenCV with YOLOv8 simplifies in a nutshell control features as well as increases the efficiency as they provide a unified platform for image processing and object detection. The user-friendly nature and high-quality accompanying documentation of OpenCV libraries allow users to implement expensive image processing algorithms more quickly in one detection framework.

Building on the usage of YOLOv8 as well as OpenCV, machine learning algorithms are a key to the system's ability to learn and confine to varying circumstances. Byte by byte training the YOLOv8 algorithm will raise its power to distinguish and categorize objects by data with diversity exposure and optimization algorithms through instructions.(Arkadiusz & Biel, n.d.)

In general, through the use of YOLOv8, OpenCV, and machine learning in our development, Otoraya Autonomous Vehicle camera system would end up having a strong and advanced functionality. Such a system will enable the car to understand the surroundings really well and therefore, it would make the right decisions and the navigation will be safe and convenient in the tricky and complex urban road system.

Figure 2.9 Algorithm for object detection (Arkadiusz & Biel, n.d.)

2.2.1 How YOLOv8 works

The YOLOv8 work flow starts off with segmenting the input image into regions or cells. By using the neural network every each cell it predicts bounding boxes as the spatial envelope of the objects and confidence scores for these predictions too. Along this, YOLOv8 uses the class probability belonging to each bounding box to determine the probable presence of particular object classes. YOLOv8 deterministically predicts the labelling by stacking various different prediction models. Accurate identification and localization of the objects are the result of this comprehensive prediction output.

The YOLOv8 has an anchor box function which acts as the initial bounding boxes' pre-defined shapes that projects the object bounding boxes. These output boxes are what allow the network to learn and predict better quality bounding boxes that can move their positions and sizes according to the positions and layouts of different objects in an image. Based on this way, the technique is not only weeding out errors, but also improving the accuracy when it comes to objects that are not of the same size or aspect ratios.(Arkadiusz & Biel, n.d.)

Conversely, OpenCV deals with Computer Vision tasks as a library which offers a vast number of features for image and video processing. With all these tools, it becomes easier to detect, track, and analyze various patterns. Opencv has been empowering different image manipulation task which include image processing, feature detection, object tracking, and other tasks based on machine learning. It is a

B. Mask Dataset Results

Fig. 3. Detection of mask on Mask Detection Datasets

Table 2. Results after training dataset-MASK

Input Image	Test		Results		
	Training	Testing	Precision	Recall	Detecting Speed
5237	4190	1047	66.5.%	62.2%	0.02 Sec/image

Figure 2.10 Detection Test (Motwani & S, 2023)

complete package that provides SDKs and algorithms which lets us do intricate and high-level computer vision tasks with relative ease. (Motwani & S, 2023)

Integrating YOLOv8 with OpenCV involves several steps:

1. Preprocessing: YOLOv8 object detection model would be initially preceded by preprocessing step such as resizing, normalization, or color model conversion. OpenCV helps in processing images in an effective way by utilizing functions for these steps.

2. Object Detection: YOLOv8 can be applied to identically prepare the input images. Next, object detection algorithm will be used to detect the target. Through the use of YOLOv8 pretrained models as well as models that are custom trained on specific databases, we are able to detect and localize objects in images, hence.

3. Post-processing: After utilizing the YOLOv8 to get object detection results, the mention of the existence of the following post-processing stages would become apparent and the need for these steps to filter and sort the detected objects would become undeniable. OpenCV comes with the functionalities for e.g. NMS, which allows to eliminate of the unneeded bounding boxes and improve the precision of the detections.

4. Visualization and Analysis: OpenCV can be utilized to show the ouput of object detection alghorthms, where bounding boxes and class labels are overlayed as a view of the input images or video streams. In addition, OpenCV gives functions for other uses beyond these like counting objects, measuring distances, or tracking objects throughout frames.(Motwani & S, 2023)

5. Integration with Applications: Subsequently, we can work out the scheme of the join YOLOv8 and OpenCV pipeline into our applications or frameworks. This integration could include providing user interfaces, creating CHIPs for external hardware devices, and progressive deployment to production lines.

The joint use of the YOLOv8 delineations and OpenCV in tracking, object recognition, and machine learning enables us to build powerful computer vision systems suitable for real-world applications. There is an immense chance of incorporating these technologies which thus brings flexibility, scalability, and performance competencies, therefore, can this suit a multitude of use-cases in multiple industries such as safety, autonomous transport, retail, and healthcare.

2.2.2 Comparison of YOLO versions

The original YOLO model and its subsequent versions are a collection of advanced neural network designs that aimed to improve on previous iterations in terms of performance, accuracy, and ease of use. Among all the versions, the first one, YOLOv1 by Joseph Redmon, proposed in 2015, introduced the real-time object detection as a single regression task. It evaluated bounding boxes and class probabilities of a full image at the same time. However, the model YOLOv1 lacked effectiveness in detecting small objects and localizing them accurately, primarily because of its coarse grid cell arrangement.(*Yolo V8: A Deep Dive Into Its Advanced Functions and New Features | by Mujtaba Raza | Medium*, n.d.)

This second version of You Only Look Once was developed in 2016 and was named YOLOv2, or YOLO9000, which fixed most of the drawbacks of the initial algorithm. It adopted mbox where improved the object detection on different scales and it was helped by a higher resolution classifier and the introduction of batch normalization. These improvements increased the speed, and also the accuracy of the system making YOLOv2 capable to detect 9132 object categories using the data from both ImageNet and COCO datasets.

In 2018, YOLOv3 brought further enhancements by utilizing a deeper network, Darknet-53, and incorporating a feature pyramid network for better detection of small objects. It offered three different scales of predictions, significantly improving performance on objects of varying sizes. YOLOv3 achieved a notable boost in accuracy while maintaining real-time speed, making it a popular choice for various applications.

Another new model, YOLOv4, was released in 2020; it incorporated several enhancements to the structure and training process. It incorporated techniques including Mish activation, Cross-Stage Partial connections (CSP), and spatial pyramid pooling (SPP). These improvements made YOLOv4 faster and more accurate that set the model to be one of the premiere models for real-time detection.

Even though there has been some confusion and debate about the naming and the history of YOLOv5, it really did become popular due to its simple usage and effectiveness. YOLOv5 was implementation with PyTorch and can be easily used as a base for object detection, including pre-trained models, data augmentation, and utilities for post-processing. Due to its focus on simple and clear code, competitive accuracy to C and faster compiling and linking it was the go-to choice for most developers.

YOLOv6 and YOLOv7 went further on developing YOLO family by paying more attention to the optimization of the backbone network and the training framework and the balance of speed and accuracy. Specifically, YOLOv7 was designed to offer better performance on resource-constrained devices such as embedded and edge AI systems.

The latest in this series is YOLOv8 which is a major advancement in the YOLO paradigm. They implemented sophisticated deep learning algorithms and improvements; hence, it remains the best tool to use in today's object detection problems. YOLOv8 uses improved backbone and head design, including modules such as EffectiveNet and transformer layers, which are applied to improve feature extraction and multiscale detection. Such improvements allow for a more accurate performance on complex datasets.

In training, YOLOv8 also employed the best practices such as various augmentation methods, learning rate schedule, and large-batch training. These innovations lead to the creation of a model with higher generalization capabilities on unseen data. In addition, YOLOv8 has a great improvement in terms of accuracy and time since it has a high score for real-time applications. Compared to the earlier versions of YOLO, it has a better map but has low latency, which is essential for the tasks requiring real-time detection.

Furthermore, YOLOv8 focuses on the usability aspect and provides well-documented APIs, clear documentation, and start points that minimize the difficulty of implementation. This makes it available to use by both researchers and other practitioners wishing to use the model. YOLOv8 is also tuned for a broad spectrum of devices the most effective to the GPUs, and the least effective to the edge devices with the least computational power. This versatility put YOLOv8 in a position where it can be used across different platforms, including big data, cloud, edge and end users. The advancements in the series of YOLOs have all played a part in enhancing real- time object detection, and YOLOv8 is the ultimate development of these systems. YOLOv8 is thus more suitable for current and future object detection tasks because it is architecturally improved and has better training methods and performance as compared to other methods. For real-time objects' detection using autonomous vehicles, smart cities, or any other applications, YOLOv8 holds the code accuracy, speed, and reliability that the contemporary world requires.

2.2.3 Our development using YOLOv8

Once we have successfully integrated YOLOv8 and OpenCV, our development process will focus on training the system to detect specific objects using depth cameras. The first step entails understanding how to transfer the data detected by YOLOv8 into actionable values that can control the autonomous vehicle's movements. This involves mapping the detected objects to corresponding commands for the car, such as turning right or left, moving backward or forward, and adjusting speed based on the situation.

To achieve this level of control, only can be grasped through depth camera for distance measure to detected items. By incorporating the data that is processed through our system is one strategy to make the object detection more accurate and logical hence the vehicle can make appropriate decisions. The depth camera will give crucial spatial data, which will supplement YOLOv8 object detection of 3D and make the system to enable recognition not only of the presence of vehicles but their proximity to the vehicle.

Along with the depth cameras, which will be in our system, the LiDAR (Light Detection and Ranging) technology, integrated with ROS (Robot Operating System) as well, will again improve the object detection and command capabilities. Topography can be reconstructed for its topology using LiDAR; nonetheless, 3D/illumination of the vehicle's surroundings provides comprehensive view of the surrounding of the vehicle. This combined result of YOLOv8's object detection module with the LiDAR data renders the object detection and localization process hi-resolution and fine-tuned. This multi-modal sensor fusion approach ensures robust performance across various environmental conditions and scenarios.

The integration of depth cameras, LiDAR, YOLOv8, and OpenCV forms the backbone of our autonomous vehicle control system. Via the provision of intense training and optimization, we aim to accomplish a smart system of high-tech nature, that can autonomously work and travel without a driver in the environment of great complexity and without any human intervention. Through utilizing smart approaches like sensor fusion mechanisms our goal is to surpass all limits of autonomy thus maintain high standards of security and reliability on road to do research and development.

2.2.4 HuskyLens AI Camera

HuskyLens AI Camera is an advanced and easy to use AI vision sensor for education and prototyping in robots and machine vision systems. This is particularly due to its Object, Line, Color, and Face detection and recognition capabilities via Machine Learning algorithms all implemented onboard the respective device. Another strength based on the manual is that HuskyLens offers an easy-to-use interface; the gadget does not need additional computational components for its main operations. This makes it very easy for students, learners, and budding developers who just want to incorporate simple artificial intelligence tasks in their programs without having to code them out.

HuskyLens can be used for multiple machine learning tasks such as, object tracking, face recognition and line following. As these capabilities are pre-built, using the camera, users can easily train the system by pointing it at objects or features that need identification. This type of design comprises a small display screen where the user experiences visual feedback in the course of learning thus making a full touch interface learning process very interactive. In that way, users can confirm exactly what the camera is capturing and operate it directly at the device, enabling more efficient working process and the absence of the necessity to program the external devices repeatedly.

The camera is compatible seamlessly with famous microcontrollers including Arduino, Raspberry Pi, or ESP32 through the I2C standard. This compatibility made it possible to incorporate it into different projects, which might make it easy to use when teaching robotics and AI. It can be coded in Arduino IDE or Python language, may fit in applications on most platforms, and is suitable for learners and experts. This versatility make it suitable for DIY robotics applications, self-driving cars and as an educational platforms for machine vision and introductory level artificial intelligence.

Possibilities of applying the HuskyLens AI Camera for educational programs, including the KTP, are numerous. Because of the uncomplicated interface and layout, the application presents significantly decreased threshold for knowledge about AI and machine vision; students can start with real practical examples of dealing with object recognition and tracking immediately using such an application without deep programming skills. Such an approach assists in creating a fun knowledge-delivery method in which students can test how unique AI technologies work and apply them within actual-life contexts. Introducing this technology in the schemes of constructing autonomous cars allows students to study the operation of vision-based systems and analyze the prospects for creating features, such as obstacle and lane detection. The HuskyLens is therefore an ideal intermediary between book learning and field experience, so makes it an ideal tool for any educational project based on Robotics and Artificial Intelligence.

2.3 Develop 1/10 simulated Advanced track

Creating a 1/10 scale Advanced Track on simulators will perform different tasks of the same kind, and altogether they will work to achieve the general aim of innovations and progress in the urban development projects and autonomous vehicles technologies. Building a scaled-down track is a building process which is dictated by the need to customize the surroundings for the Autonomous Vehicles. With the down-scaling, we are able to make a testing environment that is more manageable and easily controlled in which the activities can be simply reproduced while avoiding the risks and complications of reality.

The biggest factor in scale is size, so the main factor for building 1/10 scale would be its practicality. Through a small track model development, we are able to speed up processes, save resources and hence finally obtain a whole plan of a big track. In addition, the compact design is a great benefit for us as it allows for testing a variety of layouts and configurations for self-driving vehicles with accuracy and rapid prototyping, enabling the optimal training and evaluation of algorithms and design formats for autonomous vehicles.

Figure 2.11 Masdar Track in UAE (*Inside Masdar Track, the $22bn Eco-Project in the Arabian Desert*, n.d.)

The most powerful cities of the future are those with a vision that is bigger than providing a place to live and is aimed at stimulating economic growth, development into a new technological era. By issuing an idea to the best urban infrastructure that assist mainly in countries like Malaysia, we clear the path for innovation and progress in both local and international levels. Innovation-oriented values of futuristic cities are a primary cause for the growth of high technologies and environmental friendliness that are critical for the most countries' leadership in the digital world and getting maximum advantage on an international scale.

Iconic cities that have been enhanced with technological advances on the global level, such as the Singaporean Smart Nation initiative and the futuristic fiscal projects of the United Arab Emirates (UAE), showcase the power of integrating technology with the urban fabric. This is achieved with the application of smart sensors, AI- driven analytics, and creative strategies for urban planning which improves all these key factors of performance; resource optimization, sustainability, and the residents experience satisfaction.

The main focus in the building of an Advanced track is the incorporation of YOLOv8 technology- a highly advanced flow of objects that detects objects at real-life-time situations. Using our customized YOLOv8 and integrating most elaborate databases, we seek to equip the track with a more intelligent, safe and responsive ecosystem. The success of YOLOv8 in localizing and categorizing objects precisely would guarantee its position as a crucial component in a myriad smart track applications, which include traffic regulation and pedestrian safety to name a few, environmental monitoring and emergency responses.

At last, our Advanced track scale 1/10 construct puts together multiple things focused on the innovation of technology, an expertise of track planning and objectives of economic development. We plan to achieve this in some way or another through collaboration and initiatives in a strategic sequence that will make the blueprint for the cities of the future, where the cutting-edge technology and the sustainable urban living coexist together, and this way, the prosperity and well-being of the coming generations will be realized.

2.3.1 What is new/ benefits

The vision of an advanced track or smart track is rapidly revolutionizing urban realities around the world through the use of modern technologies to improve the performance, resilience, and livability of cities. Below are four examples of advanced cities that can be used to demonstrate how these technologies are applied and what their impact on the track residents is. It is not a secret that Singapore is one of the most developed cities in the world because of their multiple applications of smart technologies. For example, the track-state has one of the most advanced IoT networks and uses an IoT network to equip sensors and cameras throughout the track to monitor and track information such as traffic flow, air quality, public safety, and more. It is processed in real-time to create improved management and services for the track.

Technology used: IoT Sensors: Singapore makes use of a massive system of IoT devices that collect information concerning the ecological conditions, traffic, and public facilities. These sensors help the track control resources. Artificial Intelligence (AI): Machine learning technologies are used to analyze the gathered data and provide predictions that are useful in establishing major track operation strategies.

Figure 2.12 Singapore advanced track (*Singapore Is a Smart Track: Learn Why - We Build Value*, n.d.)

Autonomous Vehicles (AVs): Singapore has been a pioneer in using AVs and the country has experimented with them to try and solve their traffic congestion and transport problems. Smart Grids: The track applies the smart grid to ensure the maximization of the reliability and efficiency of the electrical supply that it is able to provide and the integration of renewable energy and energy distribution.

The benefits are: Improved Traffic Management: Traffic information monitoring and analysis can be quite effective in alleviating traffic congestion. Enhanced Public Safety: Surveillance and monitoring systems that use artificial intelligence as the underlying technology can help to enhance public safety. Sustainable Energy Use: Smart grids are efficient in eliminating energy waste and saving costs linked with sustainable energy consumption. Better Public Services: Predictive analytics improve the performance and flexibility of public sector service provision. Barcelona, Spain: Smart cities for urban growth and human growth. The track has also invested in the smart lighting, smart parking, and Internet of Things and big data based waste management.

Technologies used: Smart Lighting: The track has installed responsive lighting systems that take into account the number of pedestrians and vehicles on the roads; therefore, the track is able to reduce energy consumption and light pollution. Smart

Figure 2.13 Barcelona, Spain advanced track (*Barcelona Smart Track: Most Remarkable Example of Implementation*, n.d.)

Parking Systems: It works using sensors and mobile applications that direct drivers through available parking space meaning that they will not have to spend a lot of time searching for parking and the traffic will also be reduced. IoT-Enabled Waste Management: This device is aimed at detecting waste undertaken in bins as well as taking account of collection routes and schedules.

Benefits: Energy Savings: The adaptive lighting systems have been found to have significant energy saving and light pollution impacts. Reduced Traffic Congestion: Managing parking in the most effective way reduces traffic and helps to eliminate the danger of the additional emission of carbon dioxide. Optimized Waste Collection: Using data to plan collection routes improves the efficiency of waste collection and service and lowers costs. Dubai, UAE is often referred to as one of the most dynamic cities in the world that is keen on implementing smart track projects using advanced technologies. Smart track application is envisioned by the track as a global smart track with smart governance, smart economy and smart living.

Technologies used:Blockchain Technology: One of the ways in which blockchain technology is put to use in Dubai is for the transparency and security of government transactions, with a reduction in fraud as well as increased trust in government services. Artificial Intelligence (AI) and Machine Learning: These technologies are used to accelerate and improve public services; they enable the provision of information to citizens more rapidly and precisely. Autonomous Drones:

Figure 2.14 Dubai, UAE advanced track (*Smart Cities World - Cultural Space - Smart Track Futures Showcased in Dubai*, n.d.)

Drones are employed in various capacities including espionage and delivery services hence showcasing efficiency and security. Smart Meters and Grids: They help to conserve energy and water thus making it is less wasteful and ecologically sustainable.

Benefits: Increased Government Transparency: Blockchain technology protects confidential information and provides transparency in the activities that governments undertake. Improved Public Services: Machine learning and artificial intelligence boost the effectiveness of public service provision. Enhanced Security and Logistics: Drones are autonomous and use 24/7 surveillance and logistical operations to help the security of the track. Optimized Resource Usage: IOT supports in reducing energy and water wastage for sustainable consumption. Smart cities in Amsterdam,Netherlands are characterized by concerns in sustainability, mobility as well as concerns in data. The track has carried out many pilot initiatives aimed at finding useful solutions for the residents and the climate change problem.

Technologies used: Smart Grid Technology: The use of smart grid technology makes Amsterdam a proud track since it enables it to manage more effectively the flow

Figure 2.15 Amsterdam,Netherlands advanced track (*Smart Track Amsterdam - About Smart Cities®*, n.d.)

and consumption of energy and the way in which renewable energy is introduced into the grid. IoT Sensors: These sensors track environmental and infrastructure changes to supply information relevant for the cities' administration. Electric Vehicle (EV) Charging Stations: Combining renewable energy sources these stations promote usage of electric cars decreasing the reliance on oil. Open Data Platforms: Amsterdam allows data access and sharing about urban cities and services so that citizens and entrepreneurs can participate in the development of the track.

Benefits: Enhanced Energy Efficiency: Smart grids and electric vehicle charging facilities also lowers the carbon footprint and promote sustainable energy. Improved Environmental Health: Recycling ensures high levels of safety in the surrounding and a high standard of air quality. Empowered Citizens: Data public service facilitates the citizens' involvement in the development of the track as a part of society. Sustainable Urban Mobility: The EV infrastructure will help in saving the world from fossil fuels and also helps in advancing clean transport.

Summary Smart cities utilize a combination of technologies like the IoT, AI, smart grids, and blockchain to make their cities more effective, sustainable, and occupiable. The practices in Singapore, Barcelona, Dubai, and Amsterdam are evidence that the usage of these technologies can bring crucial advantages. Some of

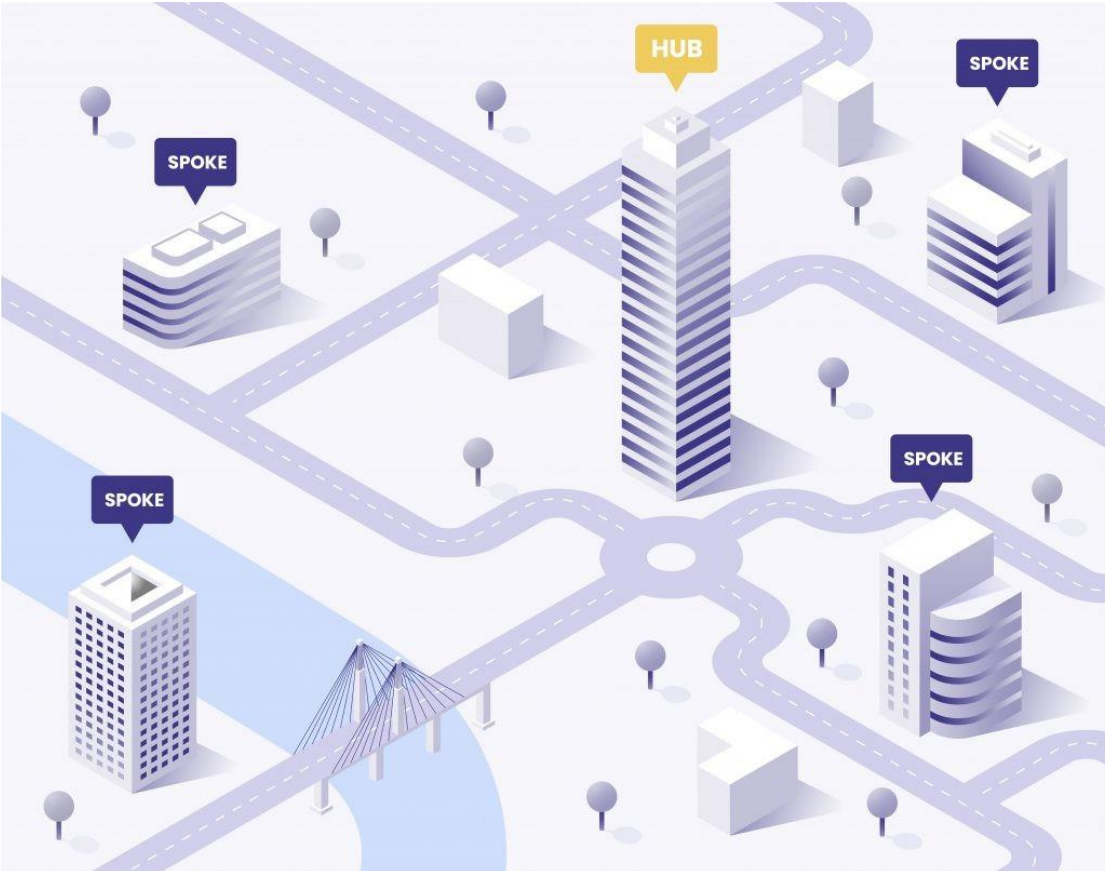


Figure 2.16 hub-and-spoke model (*Hub-and-Spoke Office Model. 4 Reasons Why It Matters - ShareSpace Blog*, n.d.)

the advantages are better improvement of citizens' security, more effective utilization of resources, advanced public services and improved standard of living. As cities in different parts of the globe keep embracing and perfecting these technologies, the prospect of building smart and dynamic trackscapes becomes more and more significant in defining the future of urban living will also define future cities.

2.3.2 Design

When planning a smart track, it is possible to use several design frameworks for the track structure that will be conducive to progressing both the track's infrastructure and population. Here, we will explore four design approaches: known as the Hub-and-Spike model, the Cluster model, the Linear model, and the Concentric model. All the design options listed above have their strengths, and we will pinpoint the best option for an advanced track. Hub-and-Spoke model is a focus on one central point within the

Figure 2.17 The Cluster model (*The Six Block Clusters and the Four Street Clusters. The Different... | Download Scientific Diagram*, n.d.)

track (for instance, the track center) with several branches in its vicinity. This central circle is where business, the government and cultures thrives while the other circles are residential and local commerce. This model is suitable for transport and access because all roads intersect in the middle producing easy and convenient access routes. It also enables efficient centralization of service and resource delivery management by the organizations. The Cluster model breaks the track into several clusters which are self-contained with clusters of residential, business and leisure areas. The transport and communication networks link the clusters. They earmark development within given regions and limit the over concentration of activities within the core business districts. It fosters people grouping in clusters and provides protection from the track-scale adverse events as each area can work individually if needed. The Linear model arranges the track in a linear pattern, often along a major transportation corridor. This design is particularly effective for cities with geographical constraints such as mountains or coastlines. The linear model ensures even development along the corridor and minimizes urban sprawl. It promotes efficient public transport systems and easy access to amenities along the linear path. The Concentric model consists of multiple rings of development radiating out from a central point. The innermost ring typically houses the central business district, surrounded by residential and commercial rings, with industrial and green belts on the outskirts. This model allows for organized growth

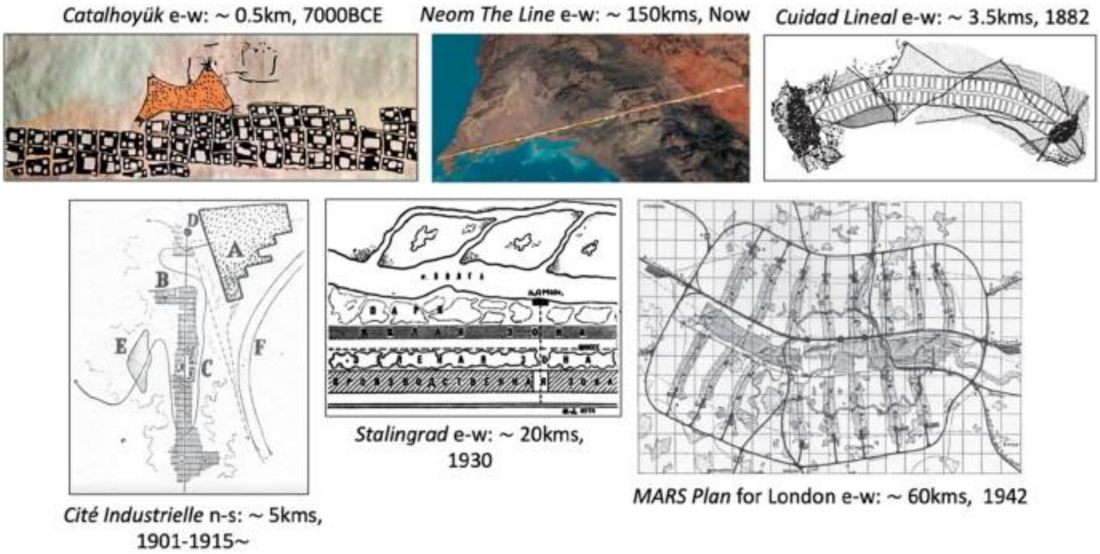


Figure 2.18 The Linear model (Batty, 2022)

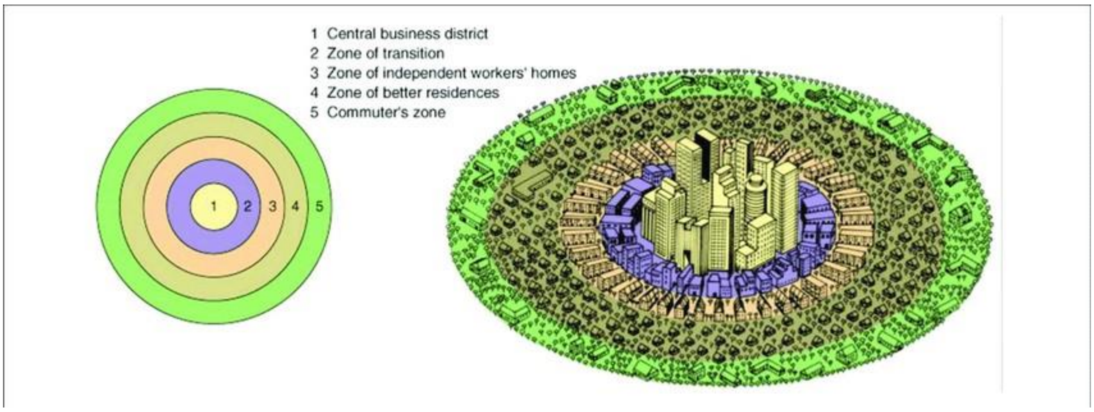


Figure 2.19 The Concentric model
(| *Graphic Illustration of Burgess' Concentric Zone Model. Nb: Zones 4... | Download Scientific Diagram*, n.d.)

and clear zoning, facilitating systematic urban planning and expansion. It supports a balanced distribution of resources and services across the track.

Based on this analysis of the four models, Cluster is the most appropriate model to be used in the advancement of an efficient track as per the sustainable track agenda, resilience and constructive community. Another aspect of the Cluster model is that the various clusters or neighborhoods created undergo development so that they are provided with basic facilities such as school, hospital, shops and places of entertainment. It minimizes the chances that people have to travel long distances in search of jobs thus decreasing traffic hazards. The model offers significant boosts for local communities since development is encouraged within clusters. Stake holders are likely to interact with other people in their neighborhoods leading to increase social relations and interactivity thus developing a good neighborhood spirit. Besides, each of the clusters can run autonomously, which makes it possible to reduce the vulnerabilities associated with any disturbances. For instance, if one part of the track has fallen short of infrastructure or disaster, the other clusters are already able to work and reduce a burden on the track. Interconnected services which are developed within clusters also ensure a good organization and allocation of resources. Micro-grids, recycling, and water treatment are easier to monitor and control in confined spaces and regions.

To put into operation the Cluster model in an advanced track, different technological and infrastructural factors have to be combined. Smart transportation networks like, autonomous electric vehicles, bike sharing, and effective public transports connecting each cluster must be established. These networks should be run by live data intelligence so that the flow of traffic can be handled well and emissions controlled well. Each cluster should have its own energy infrastructure independent from the centralized power plant and rely on renewable energy, perhaps solar panels and windmills. The concept of smart grid allows for optimally in energy distribution hence guaranteeing stability in supply. The technological needs of a track should involve IoT for managing resources in the infrastructure. Usefulness and management of its usage in smart streetlights, waste management systems and water supply networks are also possible. Install community oriented amenities that include; health facilities, learning institutions, and playing fields within those clusters. These services should ensure efficient management and deployment through technology, for instance, the use of telemeters for health centers and virtual classrooms in schools. Include as many open spaces within clusters as possible to improve the population's well-being and reduce their adverse impact on the environment. Converting building rooftops into greenery, gardens, and parks in the urban centers offer benefits such as enhanced air quality, spaces for recreation, and the support of biological diversity.

The Cluster model appears to be more practical and adaptive in the creation of smart cities, giving a central approach that is centered on locality, people, and sustainability. This model of smart technology and efficient resource management is designed to augment the track's physical structure and populace for the better. Some of the most developed cities such as Amsterdam and Singapore have integrated aspects of this model, proving its efficiency enables urban areas to develop into sustainable cities.

2.3.3 Ethical perspective

The ethical integrity of autonomous cars and advanced cities is a thing that involves complicated issues like privacy, security, and the proper use of data. Although the gathering and processing of huge sets of data are the main factors of the efficiency and the safety of these systems, the current situation demands the attention to the danger of the unauthorized access or the misuse of the sensitive information.

The main ethical problem is data privacy and security. Autonomous cars and the smart track infrastructure are built to depend on the large data collection from various sources like sensors, cameras, and others for making the right decisions and for the enhancement of the functionality. Nevertheless, the enormous amount of data, if not managed properly or leaked, could be dangerous to the privacy of the persons and public safety. Unauthorized access to personal information and real-time location data, for instance, could result in identity theft, surveillance abuse, or even physical harm.

To deal with these threats, strong data protection mechanisms have to be used in every part of data collection, storage, and transmission. Encryption, access controls, and anonymization methods are the means for protection of sensitive information and to restrict the access to it. Besides, the fact of complying with privacy regulations like the General Data Protection Regulation (GDPR) in Europe or the Personal Data Protection Act (PDPA) in Malaysia, is very important in order to meet the legal requirements and to maintain the private people's rights.

Besides, transparency and accountability are the basic principles of ethical governance in autonomous systems and smart cities. The data collectors have to be made aware of the kind of data they are collecting, the way it is being used, and the people who have access to it. Clear guidelines and rules should be laid down to control the data usage and guarantee that the data is being used with responsibility and ethics. The regular audits and oversight mechanisms will be able to monitor the compliance and find any of the possible breaches of the ethical standards.

Besides, the vital thing in the ethical consideration is the danger of data misuse or the exploitation by the bad guys. Though the current security measures are very strong, there is still a possibility of data breaches or cyberattacks that could affect the confidential data. The problem at hand needs to be solved through a broad approach which includes technical solutions and policy interventions.

A feasible resolution is the adoption of strict data retention policies that dictate the period of data storage. The automatic deletion or anonymization of the data after a certain period of time lowers the risk of long-term exposure to the data breaches. Besides, the organization of secure data storage facilities with limited access, like a separate system that can be used only by the authorized persons, can be another way of the protection of the confidential information.

Moreover, firm governance and the control mechanisms are the key to blaming the people who are in charge of data management and safety. Assigning a data protection officer or a committee that will handle data management practices and make sure that these practices are ethical can help the public to trust the authentitrack and security of the autonomous systems and the smart track infrastructure.

Finally, the ethical problems that are linked to autonomous cars and the advanced cities need to be solved in a holistic way that puts privacy, security and accountability first. Through the use of reliable data protection mechanisms, trust and accountability, and innovative solutions to the problems of data misuse we can create a base for ethical and responsible use of autonomous technologies in our globally connected world.

2.3.4 Malaysian Road Standard

In my 1/10 model track, I am keen to follow the Malaysian road standards whenever I am mapping my track so that it is more real like. Malaysia is divided into a broad network of road facilities which include expressway, arterial road, collector road and local streets and the design of these roads is done to the most standards set by the Public Works Department (JKR). The different categories of roads contain scalable parameters of the lane width, shoulder width, and the central reservation width.

For instance, the R6/U6 expressway with high-speed characteristics has a standard lane width of 3.75 meters, a median (central reservation) width of 3.5 meters, and shoulder widths of 3.0 meters at the right and 2.5 meters at the left. Collector roads (Jalan Persekutuan) as well as local streets, generally have tended to have a narrower carriageway, running at an approximate of 3.0 to 3.25 meters wide based on a variation on its type and usage.

For purposes of maintaining the realism of the model, I scaled the above dimensions by a factor of 10: This means that the actual model utilized a dimension of 1/10. This implies that if normal expressway lane width is 3.75 meters this is equivalent 0.375 meters (37.5 cm) on the model. Likewise, the 3.5 meters of central reservation is converted to 0.35 meters (35 cm) while shoulder widths of 0.3 meters (30 cm) and 0.25 meters (25 cm) are also employed. This precise scaling makes sure that the model track retains scale integrity and represents the real world design as fully as is possible.

Besides, the intersections, roundabouts, and pedestrian walkways in the model are established according to the Malaysian standards related to track traffic organization. Other features such as traffic lights, road signs and demarcation lines on the roads used in the project are also made proportionate to the dimensions on the road to make the project as close to the real world as possible.

By adhering to these standards, the model track provides an excellent educational tool, allowing students to experience road design principles firsthand. The realistic road infrastructure not only makes the autonomous vehicle testing more accurate but also helps in demonstrating traffic rules and urban planning concepts.

For reference, I consulted the" Arahan Teknik Jalan (ATJ)" manuals published by JKR, specifically the ATJ 8/86 (Geometric Design Guide) and ATJ 5/85 (Road Furniture), which outline the official specifications for Malaysian roads. (*02_Guidelines_Geometric_Standards_On_Roads_Networks_System*, n.d.)

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

Incorporation of state-of-the-art object detection system particularly YOLOv8 with physical camera and Lidar system ensures a sound perception system for self-driving cars in a smart track environment. It is aimed at promoting further advancements in accuracy, productivity, and dependability of car guidance and protection systems in such areas. This being the direction that smart track development is taking – to integrate advanced technologies to enhance physical structures, security, and the overall well-being of citizens, our project will also help add to the upswing of smart track progress to achieve safer and efficient transit means.

Smart cities are features that are intended to incorporate the use of different advanced technologies to promote effective and sustainable utilization of urban settings. These cities use sensors, data analysis, and active networks to coordinate resources and services in the most efficient way possible. Here in our smart track model, YOLOv8 will be productive on object detection and classification to help on real-time decision and control of autonomous cars. To be integrated into the camera and LiDAR based systems, YOLOv8 is aimed at improving the car's environment awareness and as a result, the safety of both vehicle occupants and road users through efficient navigation and obstacle detection.

The selection process is initiated by choosing the YOLOv8 model for its high efficiency in object detection compared to others. YOLOv8 has enhancements over the previous models in terms of accuracy and speed through future optimizations such as Efficient-Net and Transformer. These improvements allow YOLOv8 to work efficiently with difficult and various datasets as well as with real-world problems, and this is why it was chosen for our project.

The first of the two aforementioned components is the camera-based setup which uses RGB cameras of high resolution for capturing details of the environment. These images are then passed to YOLOv8 which is tasked with detecting and determining what in the scene is a pedestrian, vehicle among others. Together with LiDAR systems that enable depth sensing and the creation of a depth map of the area, it is possible to consider the situation in a broad context. These assist in decision making within the self driven car such as the change in speed, moving on to a new lane and even in avoiding an object.

Also, the demand for such integrated systems in the market is on the rise as well continuously. As the call for smart track and self-driving cars progresses, the further attempts for the creation of technologies that can provide efficient solutions on mobility continue. This need is met by using YOLOv8 for merging camera and LiDAR-based systems to offer a perception system that is capable of operation in various conditions and scenarios within the urban environment.

3.2 Autonomous Vehicle

3.2.1 Product Design Specification (PDS) for Autonomous Vehicle

Performance: Thus, the automobile is planned to be an AV- camera-sensor system for identification and control. In its simplest form, it is designed to resemble actual driving conditions in a 1/10 scale model. Key performance requirements include:

- Response Time: Below 1 second for target identification and decision making.
- Speed: A maximum of speed of 20 km/h adopted in the actual operation of the car, reduced to the speed of the 1/10 model to achieve realistic performance in a safer environment.
- Load: The vehicle must self-weight the same and have the dynamics of 1/10 model of the regular car so that the real car dynamics is emulated to the t.

Customers: The primary audience for this product includes:

Educational Institutions: Educational institutions which enroll students for STEM programs and encourage students with knowledge in other related aspects of self-driven vehicles. Students and Hobbyists: Including them in coding exercises, use of sensors and exploring activities on robotics.

Appearance/Aesthetic: The vehicle has to be attractive to draw the user's attention particularly for students. Key aesthetic considerations include:

- A small and stylish look alike of an actual car on the road.
- Lively colors and carefully written tags of parts for teaching purposes.
- Sturdy and small construction for ease of transport from one place to another.
- That contains such realistic features as scale model headlights, taillights and body lines.

Testing: Planned tests for the product include:

- Accelerated Life Testing: Imitate consistent use to assess how the product will whine after long-time utilisation.
- Fatigue Testing: Discuss the micro and macro yielding behavior under repeated loads and stress. Sensor Accuracy Testing: It has a purpose to confirm the detection and response capability of the camera.
- The use of software logs in monitoring the performances of system or components during tests in real-time.
- Visual checks for signs of wear in the physical components and checking for signs of possible damage.
- Another component of estimated costs are testing costs which include the application of laboratory equipment and tools software and personnel time estimated to range between 500RM and 1000RM based on the tests to be done.

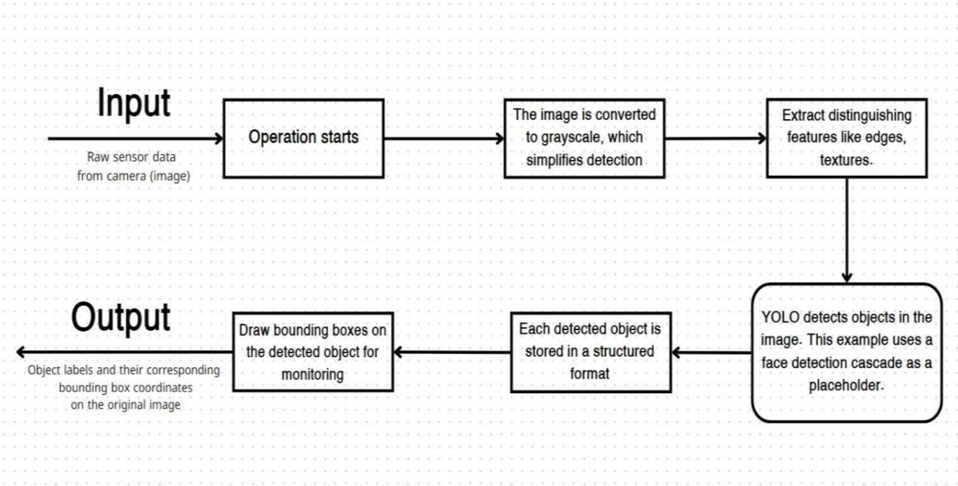


Figure 3.1 Function Decomposition

3.2.2 Proposed Method

In this specification, for the development of a highly efficient highly automated vehicle, it is required to incorporate YOLOv8, OpenCV, and possibly CNN for efficient and effective object detection and decision making. Therefore, by using all these enhanced technologies coupled with the depth camera we expect to develop a stable system which will allow for accurate navigation and to avoid all the obstacles. For this purpose, the system will be equipped with a robust processor and refined coding in order to make it efficient and functional. For efficient control, the vehicle is also going to incorporate servos The vehicle is also going to be capable of achieving a maximum speed of 50 km/h in the absence of any external intervention.

The first part of the present method is the ability to perform object detection with the help of YOLOv8. The YOLOv8 model, which popular for both its high accuracy and fast run time, is going to be used for object detection and segmentation within the environment of the vehicle. By utilizing the images coming from the RGB cameras, YOLOv8 will identify pedestrians, vehicles, and other obstacles immediately in real-time. The incorporation of OpenCV helps in detail filtering of the images, edge detection, as well as the extraction of features, which ultimately increases the effectiveness of the detection system. But if while developing we come across a situation where CNN is more efficient or has higher accuracy in the specific application, then we could go for it rather than OpenCV.

To increase the level of accuracy of the system further, let us use a depth camera. This particular piece of equipment will prove to be essential in identifying the depth of the objects that are picked up by the camera with a view to helping the vehicle make correct decisions on what to do next or how to avoid the detected objects. This depth information will be useful for accurately determining safe stopping distances, more efficient path planning as well as dynamic speed control in relation to objects nearby.

The computational requirements of the system will be processed by a powerful processor that will be able to perform the necessary computational algorithms as well as ensure real time data processing, which will be crucial for autonomous driving. The processor will also make sure that the car can quickly and efficiently sort all of the information coming from the cameras and the sensors in order to make timely and accurate decisions as well as control the actions of the vehicle.

The mechanical part of the vehicle will be controlled using servos for the purpose of turning the wheels and controlling the throttle. These servos will be managed by the AI system which will analyze the YOLOv8 information, depth camera, and other devices for the proper instructions on the next action to be taken. By integrating superior quality mechanical parts with superior artificial intelligence, the car will be able to drive through the sophisticated terrains and can travel with the speed of 50/Km per hour without the interference of human beings.

3.2.3 Installation procedure

Before embarking on the development of our autonomous vehicle system, it is important to load the following software tools and libraries on the system environment. Here is the installation guide on Visual Studio, Python, YOLOv8 from scratch:

Step 1: Install Visual Studio 1.Download Visual Studio: Visit the Visual Studio download page and download the Community edition.

2. Run the Installer: Launch the downloaded installer.

3. Select Workloads: In the installer, select the following workloads:

- Desktop development with C++: Required for building native applications.
- Python development: Required for Python integration.
- NET desktop development (optional, for broader capabilities).

4. Install: Click on the Install button to begin the installation process. This may take some time depending on your internet speed and system performance.

5. Launch Visual Studio: Once the installation is complete, open Visual Studio to ensure it is working correctly.

Step 2: Install Python

(a) Download Python: Go to the Python official website and download the latest version of Python (preferably Python 3.8 or above).

(b) Run the Installer: Launch the downloaded installer.

(c) Add Python to PATH: During the installation process, make sure to check the box that says ”Add Python to PATH”.

(d) Install: Follow the prompts to complete the installation.

(e) Verify Installation: Open a command prompt and type python –version to ensure Python is installed correctly.

Step 3: Set Up a Virtual Environment

(a) Open Command Prompt: Open a command prompt or terminal window.

(b) Navigate to Project Directory: Change the directory to where you want to set up your project, e.g., .

```
python -m venv env
```

Figure 3.2 Virtual Environment command

```sh
.\env\Scripts\activate
```

Figure 3.3 Windows

(c) Create Virtual Environment: Run the command in figure 3.2 to create a virtual environment

(d) Activate Virtual Environment: Activate the virtual environment as shown in Figure 3.3 and 3.4

```sh
source env/bin/activate
```

Figure 3.4 Mac/Linux

Step 4: Install Required Libraries

- Upgrade pip: Ensure you have the latest version of pip
- Install YOLOv8: Install the YOLOv8 library via PyTorch or another relevant package manager. Since YOLOv8 is often used through Ultralytics' repository, you can use the following command:
- Install OpenCV: Install the OpenCV library using pip:
- Install Additional Dependencies: Depending on your project needs, you might need other libraries such as NumPy, Matplotlib, or specific machine learning frameworks: pip install numpy matplotlib

```sh
pip install --upgrade pip
```

Figure 3.5 Upgrade pip

```sh
pip install ultralytics
```

Figure 3.6 Install YOLOv8

Step 5: Configure Visual Studio for Python Development

- Open Visual Studio: Launch Visual Studio.
- Create a New Project: Go to File ¿ New ¿ Project.
- Select Python Project: Choose a Python application template.
- Set Up Interpreter: Ensure that Visual Studio is using the Python interpreter from your virtual environment. Go to Tools ¿ Options ¿ Python ¿ Environment and set the correct interpreter path (.slash env slash Scripts slash python.exe).

Step 6: Test the Installation

- Create a Test Script: In Visual Studio, create a new Python script (e.g., testunderscorescript.py).
- Write Test Code: Add the following code to test if everything is installed correctly:
- Run the Script: Execute the script in Visual Studio to verify that OpenCV, PyTorch, and YOLOv8 are working correctly.

Following these steps will set up a comprehensive development environment for creating and deploying an autonomous vehicle system using YOLOv8, OpenCV, and

```sh
pip install opencv-python
```

Figure 3.7 Install OpenCV

```python
import cv2
import torch
from ultralytics import YOLO

print(f"OpenCV version: {cv2.__version__}")
print(f"PyTorch version: {torch.__version__}")
model = YOLO('yolov8s.pt')
print("YOLOv8 loaded successfully")
```

Figure 3.8 Check installation

Python. This setup ensures that all necessary tools and libraries are installed and configured properly, allowing for seamless development and testing.

3.2.4 Required component

To create the system of autonomous vehicles, several components are required. All these sub systems work together to help facilitate the mode of transport and performance of the vehicle. Here, we explain each component in detail:

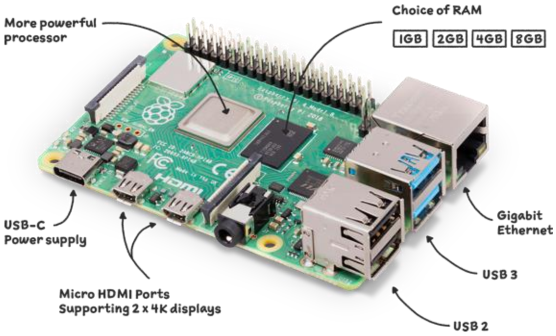


Figure 3.9 Raspberry Pi

Processor: Raspberry Pi The Raspberry Pi is a small and multi-functional single board computer that acts as the control center of our self-driving car. In particular, it is responsible for executing the software, as well as for the processing of data coming from the sensors, and the management of the motors' operations. For this project, we shall be using a Raspberry Pi 4 Model B that has a quad core Cortex- A72 CPU, 4GB RAM, and several I/O interfaces. This model offers the necessary computation and interfacing features to address real-time applications such as object detection and motors' control.

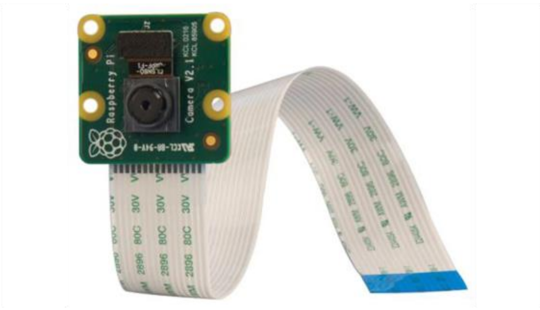


Figure 3.10 Camera Sensor

Camera Sensor The camera sensor is a significant subassembly that takes photo-realistic images for real-time object recognition. The camera used will be Raspberry Pi Camera Module V2 that has an 8-megapixel capatrack to shoot HD images as well as videos. This camera does not require any additional ports since it plugs directly into the Raspberry Pi through the CSI port, and is also very fast. For the obstacle detection and recognition to be performed in real-time, this will be undertaken along with the help of the camera and known algorithm named YOLOv8.

Servo Motor for Steering The servo motor is applied to the steering system of the vehicle so as to enable or manage its movement. For motor we will take a standard 9g micro servo motor with which we can accurately control the steering angle. The servo motor is then connected to the steering linkage of the car kit,thus allowing the Raspberry Pi to program the movements where and when the car kit should to via the data processing.

Figure 3.11 Servo Motor for Steering

Figure 3.12 DC Motor

DC Motor for Movement This motor like the name suggests is used in moving the wheels or turning the vehicle forward or reverses. As the motor of small size, we will utilize a small direct current or DC motor that is capable of offering suitable torque and sufficient speed to a car. The motor shall be managed through the motor driver hence enabling us to control the speed and direction of the vehicle.

Motor Driver The motor driver is a circuit that allows the Raspberry Pi to connect to the motors. There is a commercially available motor driver called L298N whose features enable it to drive two DC motors in an independent manner. This driver will help in controlling the rotation speed and that direction of the both the DC motor and the servo motor. This offers the required power amplification to power the motors

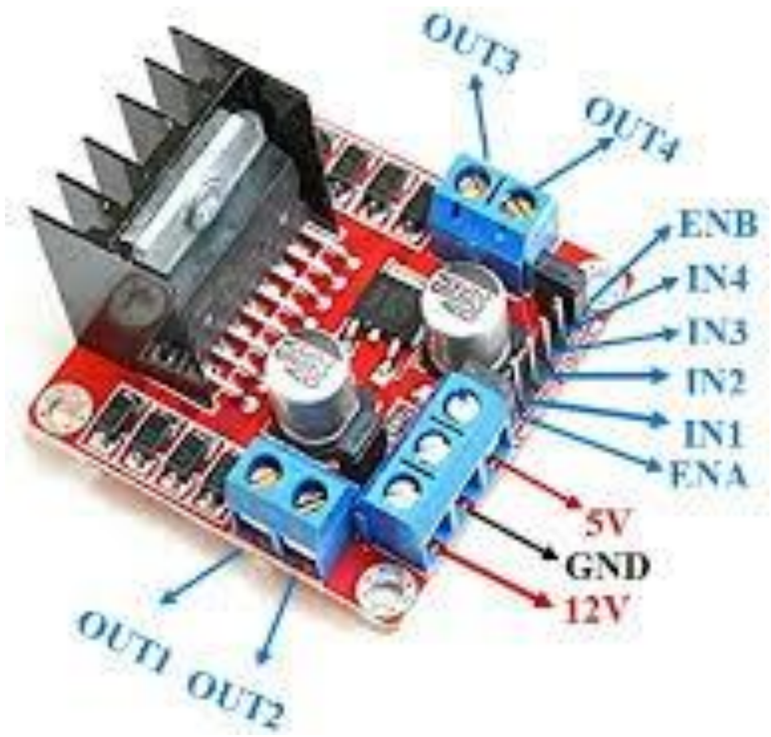


Figure 3.13 Motor Driver

on the one hand and receive PWM signals for controlling the power being supplied by the Raspberry Pi on the other hand.

Figure 3.14 Nickel Battery

Nickel Battery The nickel battery is in charge of supplying electritrack for the vehicle. A Nickel-Metal Hydride (NiMH) battery pack will be employed here since they are rechargeable and among the most dependable batteries. NiMH batteries are characterized by their stability and capatrack capable of driving forward both the motors and the Raspberry Pi. A typical 7.2V NiMH battery pack will be employed For sufficient power supply to the entire system.

Figure 3.15 Wires and Breadboard

Wires and Breadboard To connect all the components, wires and a breadboard should be used. Breadboard helps with circuit construction and experimentation as no soldering is involved and hence, a change or two can be made easily. The wires are utilized to create the connection between the Raspberry Pi, motor driver, the camera module, as well as the motors. Flexible and easy to connectors will be male to male jumper wires and male to female jumper wires.

Car Kit The car kit is the body and sandy structure of the car. This system encompasses properties like the body, wheels, axles, and manner of steering. For this project we will start using a conventional RC car model kit of 1/10 scale as it is sturdy and easily expansible. The new appearance of the car kit will be designed and added with input-output interfaces for the Raspberry Pi, camera module and motors besides the many other electronics.

Memory Card and Card Reader. Specifically, the memory card here is to store the operating system and software for the Raspberry Pi application. The microSD card that will be used should be of high speed and should have sufficient storage space; at

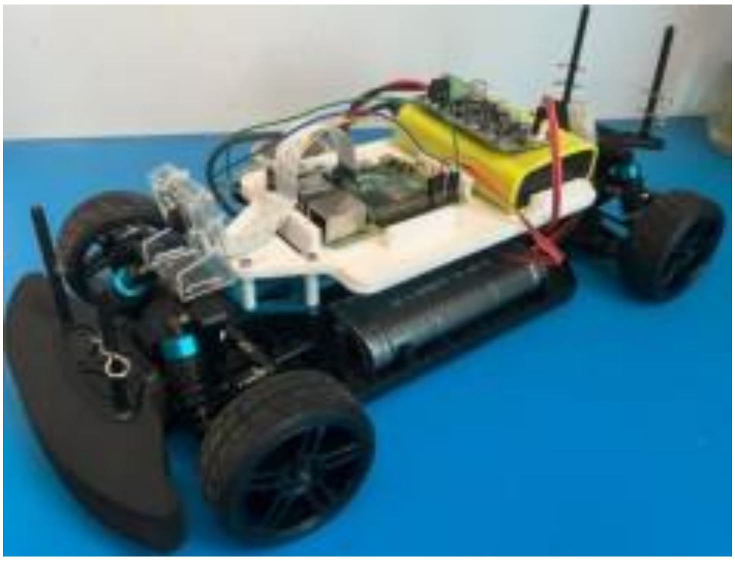


Figure 3.16 Car kit

least 32GB. Partially, it is planned to use a memory card reader in order to copy files and download the operating system on the microSD card. Also, the card reader assists in the easy updating and backing up of the system software at the same time.

USB-C Cable A USB-C cable is used for all the troubleshooting and resetting processes of the Raspberry Pi. This cable enables us to link the Raspberry Pi with a computer to allow us access to the CUI, the files as well as debuggers. The USB-C port is also used for charging when the car is not operating with the battery pack and for easy development and testing.

When incorporated, the mentioned components will make it possible to design an autonomous vehicle that can locate its environment and move through its surroundings using vision and motor control. The main computational element is provided by Raspberry Pi along with the camera sensor, motors, and the motor driver to allow the car 'see' and navigate. These includes the power supply, the connectivity tools and structural members which contribute towards the effectiveness of the system.

Figure 3.17 A memory card reader

Figure 3.18 USB-C Cable

3.2.5 Wiring and setup

Setting up your Raspberry Pi involves preparing the hardware, installing the Raspbian OS, and configuring it to be ready for use. Follow these steps for a smooth setup process:

1. Preparing the Hardware

(a) Gather Required Components:

- Raspberry Pi 4 Model B
- MicroSD card (32GB or larger)

- MicroSD card reader
- USB-C power supply
- Raspberry Pi Camera Module V2
- HDMI cable (optional, for initial setup)
- Monitor and keyboard (optional, for initial setup)

(b) Assemble the Raspberry Pi:

- Place the Raspberry Pi into the case (if you have one).
- Connect the monitor using the HDMI cable.
- Connect the keyboard and mouse to the USB ports.
- Insert the microSD card into the microSD card slot on the Raspberry Pi.

2. Downloading and Installing Raspbian OS

(a) Download Raspberry Pi Imager:

- Visit the official Raspberry Pi website: Raspberry Pi Downloads.
- Download the Raspberry Pi Imager for your operating system (Windows, macOS, or Linux).
- Install the Raspberry Pi Imager on your computer.

(b) Prepare the MicroSD Card:

- Insert the microSD card into the card reader and connect it to your computer.
- Open the Raspberry Pi Imager.

3. Setting Up the Raspberry Pi

(a) Initial Boot:

- nsert the microSD card into the microSD card slot on the Raspberry Pi.

- Connect the USB-C power supply to the Raspberry Pi and plug it into a power outlet.
- The Raspberry Pi will boot up, and you should see the Raspbian desktop on your monitor.

(b) Initial Configuration:

- On the first boot, the Raspberry Pi will prompt you to configure some initial settings:
- Select your country, language, and timezone.
- Create a user account and password.
- Connect to your Wi-Fi network or configure a wired network connection.
- Update the system software if prompted.

(c) Enable SSH (Optional):

- If you plan to access your Raspberry Pi remotely, you need to enable SSH:
- Open the Raspberry Pi configuration tool by clicking the Raspberry Pi icon in the top-left corner of the desktop, then navigate to ”Preferences” ¿ ”Raspberry Pi Configuration”.
- Go to the ”Interfaces” tab and enable SSH.
- Click ”OK” to apply the changes.

(d) Install Additional Software (Optional):

- You can install additional software packages as needed using the terminal:
- Open the terminal by clicking the terminal icon on the desktop.
- Update the package list and upgrade installed packages as shown in figure 3.18
- Install additional software using apt install as shown in figure 3.19

(e) Configure the Camera Module (Optional):

```sh
sudo apt update
sudo apt upgrade
```

Figure 3.19 Update the package list

```sh
sudo apt install package-name
```

Figure 3.20 apt install

- If you are using the Raspberry Pi Camera Module, enable it in the configuration tool:
- Open the Raspberry Pi configuration tool from the ”Preferences” menu.
- Go to the ”Interfaces” tab and enable the camera.
- Click ”OK” and reboot the Raspberry Pi to apply the changes.

4. Finalizing the Setup

(a) Configure Auto-Login (Optional):

- To configure the Raspberry Pi to log in automatically:
- Open the Raspberry Pi configuration tool.
- Go to the ”System” tab and set ”Auto-login” to ”Enabled”.
- Click ”OK” to apply the changes.

(b) Set Up Remote Desktop (Optional):

- If you want to access your Raspberry Pi’s desktop remotely, you can install and configure VNC:
  - Open the terminal and install the VNC server as shown in figure 3.20
  - Enable VNC in the Raspberry Pi configuration tool under the ”Interfaces” tab.

```sh
sudo apt install realvnc-vnc-server
```

Figure 3.21 Set Up Remote Desktop

(c) Backup Your Setup (Optional):

- Once your Raspberry Pi is fully configured, it's a good idea to back up your microSD card:
  – Shut down the Raspberry Pi and remove the microSD card.
  – Use a disk imaging tool (such as Win32DiskImager for Windows or dd command on Linux) to create an image of your microSD card.

By following these detailed steps, you will have a fully operational Raspberry Pi with Raspbian OS, ready for various applications, including your autonomous vehicle project.

Setting up the Raspberry Pi for our autonomous vehicle system involves several key steps. These include preparing the hardware, installing the necessary software, and configuring the system to work with the components. Here is a detailed step-by-step guide to setting up the Raspberry Pi:

1. Preparing the Raspberry Pi

(a) Assemble the Raspberry Pi:

- Insert the microSD card into the card reader and connect it to your computer.
- Open the Raspberry Pi Imager, select the Raspberry Pi OS (32-bit) as the operating system, and choose your microSD card as the storage device.
- Click ”Write” to flash the operating system onto the microSD card.

(b) Insert the MicroSD Card:

- Once the flashing process is complete, safely eject the microSD card from your computer.
- Insert the microSD card into the microSD card slot on the underside of the Raspberry Pi.

(c) Connect the Camera Module:

- Attach the camera module to the CSI port on the Raspberry Pi. Make sure to lift the plastic clip on the CSI port, insert the camera ribbon cable with the blue side facing the Ethernet port, and press the clip back down to secure it.

(d) Connect the Power Supply:

- Connect the USB-C power supply to the Raspberry Pi and plug it into a power outlet.

(e) Initial Setup (Optional):

- If you are setting up the Raspberry Pi for the first time, connect it to a monitor using an HDMI cable and plug in a keyboard. This will allow you to perform the initial setup directly on the Raspberry Pi.

2. setting up Necessary Software

(a) Connect to Wi-Fi:

- If using a monitor, you can connect to Wi-Fi through the graphical interface. If headless, create a file named wpa...supplicant.conf in the root directory of the microSD card with the following content as shown in figure 3.21
- Replace YOURCOUNTRYCODE, YOURSSID, and YOURPASSWORD with your Wi-Fi network details.

(b) Update the System:

```makefile
country=YOUR_COUNTRY_CODE
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
  ssid="YOUR_SSID"
  psk="YOUR_PASSWORD"
}
```

Figure 3.22 Connect to Wi-Fi

```sh
sudo apt update
sudo apt upgrade
```

Figure 3.23 Update the System

- Once the Raspberry Pi is connected to the internet, open a terminal (or SSH into it) and update the system packages by running the commands as shown in figure 3.22

(c) Install Python and Required Libraries:

- Install Python and pip, if not already installed as shown in figure 3.23
- Install necessary Python libraries, including OpenCV and the YOLOv8 library as shown in figure 3.24

3. Configuring the System

(a) Enable the Camera Interface:

- Run sudo raspi-config to open the Raspberry Pi configuration tool.
- Navigate to Interfacing Options and enable the camera interface.

```sh
sudo apt install python3 python3-pip
```

Figure 3.24 Install Python

```sh
pip3 install opencv-python-headless
pip3 install ultralytics
```

Figure 3.25 OpenCV and the YOLOv8

```sh
raspistill -o test.jpg
```

Figure 3.26 Test the Camera

- Reboot the Raspberry Pi to apply the changes.

(b) Test the Camera:

- Verify that the camera is working by capturing an image using the raspistill command as shown in figure 3.25

(c) Connect the Motors and Motor Driver as shown in figure 3.29 and 3.27.

- Connect the servo motor and DC motor to the motor driver as per the manufacturer’s instructions.
- Connect the motor driver to the Raspberry Pi GPIO pins, ensuring proper connections for power, ground, and control signals.

(d) Write and Deploy the Control Script:

- Write a Python script to integrate YOLOv8 with the camera feed and control the motors based on detected objects as shown in figure 3.30.
- Use OpenCV to capture images from the camera, process them with YOLOv8, and send commands to the motor driver as shown in figure 3.31.
- Deploy the script on the Raspberry Pi and run it to test the autonomous vehicle system.

(e) Fine-Tune and Optimize:

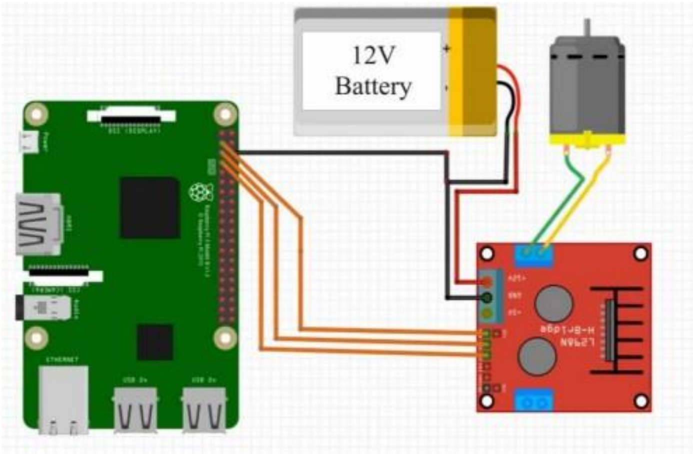


Figure 3.27 All components wiring needed

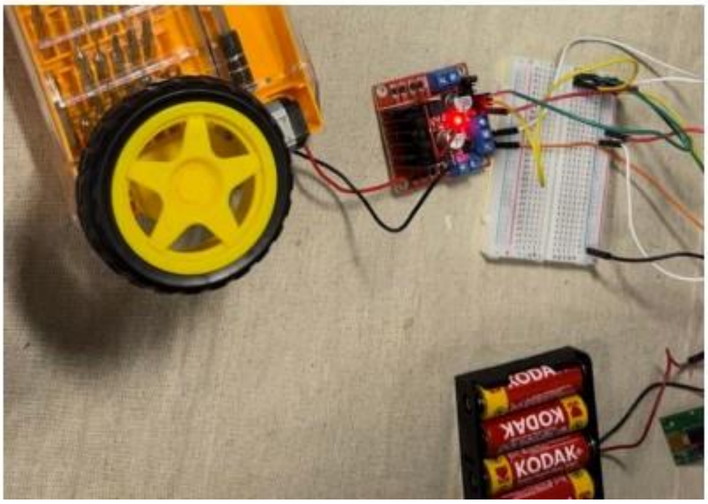


Figure 3.28 Dc motor and motor drive

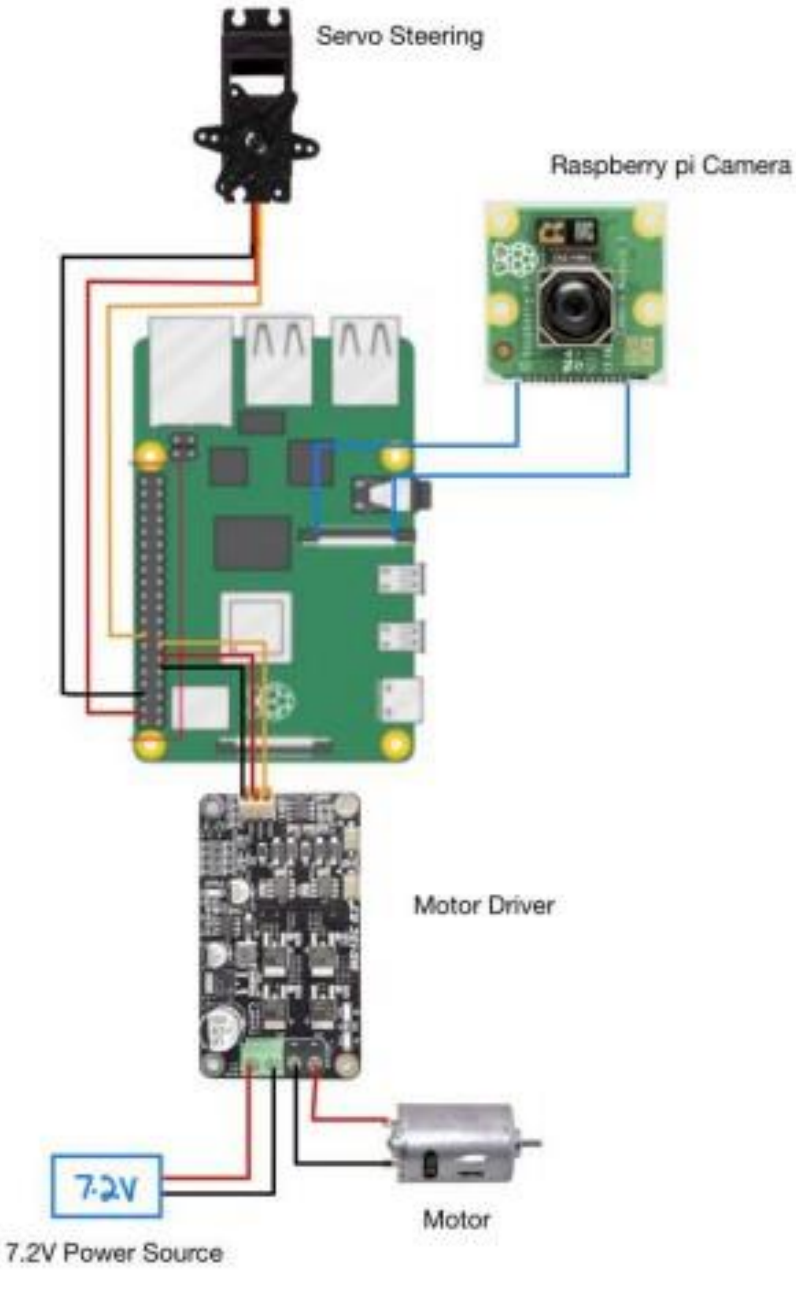


Figure 3.29 Motor drive wiring

```python
# the first thing is the class , the bounding box (the center for the first two ), the width then the hight
from ultralytics import YOLO
import os
import cv2
import numpy
import random
#results = model.train(data="config.yaml", epochs=3) #tip make sure all of ur loos going down


frame_wid = 640
frame_hyt = 480

cap = cv2.VideoCapture("car.mp4")
#model_path = os.path.join("c:/", "opencv/runs/detect/train7/weights/last.pt")
#model = YOLO(model_path)  # load a custom model
model = YOLO("weights/yolov8n.pt", "v8")

threshold = 0.5
#results = model.predict(source="0", show=True)

if not cap.isOpened() :
    print("Cannot open camera")
    exit()

while True:
    # Capture frame-by-frame
    ret, frame = cap.read()
    # if frame is read correctly ret is True

    if not ret:
        print("Can't receive frame (stream end?). Exiting ...")
        break
    results = model(frame)[0]
    cv2.imshow("ObjectDetection", frame)


    for result in results.boxes.data.tolist():
        x1, y1, x2, y2, score, class_id = result
```

Figure 3.30 Integrating open cv and Yolov8

```python
if not cap.isOpened() :
    print("Cannot open camera")
    exit()

while True:
    # Capture frame-by-frame
    ret, frame = cap.read()
    # if frame is read correctly ret is True

    if not ret:
        print("Can't receive frame (stream end?). Exiting ...")
        break
    results = model(frame)[0]
    cv2.imshow("ObjectDetection", frame)


    for result in results.boxes.data.tolist():
        x1, y1, x2, y2, score, class_id = result

        if score > threshold:
            cv2.rectangle(frame, (int(x1), int(y1)), (int(x2), int(y2)), (0, 255, 0), 4)
            cv2.putText(frame, results.names[int(class_id)].upper(), (int(x1), int(y1 - 10)),
                        cv2.FONT_HERSHEY_SIMPLEX, 1.3, (0, 255, 0), 3, cv2.LINE_AA)


    cv2.imshow("ObjectDetection", frame)


    # Terminate run when "Q" pressed
    if cv2.waitKey(1) == ord("q"):
        break

# When everything done, release the capture

cap.release()
cv2.destroyAllWindows()
```

Figure 3.31 Integrating open cv and Yolov8 second part

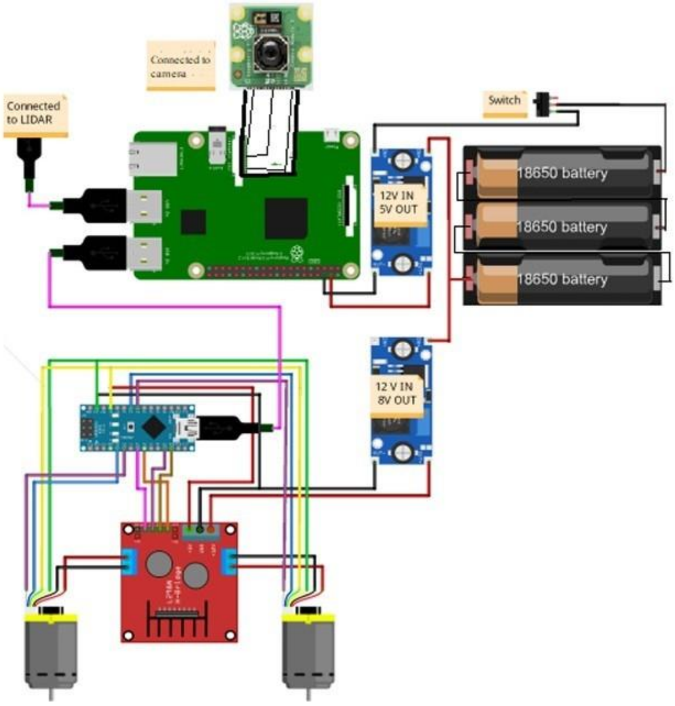


Figure 3.32 Circuit and wiring

- Continuously test and optimize the system for better performance. Adjust the object detection parameters, motor control logic, and other settings as needed.

By following these steps, you will have a fully set up Raspberry Pi system ready to control the autonomous vehicle using YOLOv8, OpenCV, and other integrated components. This setup forms the foundation for building a sophisticated and reliable autonomous vehicle capable of navigating its environment with minimal human intervention.

3.2.6 Training and machine learning

Training a custom dataset for object detection involves several steps, from downloading and labeling your dataset to configuring your training environment and running the training process. Here is a detailed guide to help you through this process:

1. Downloading the Dataset

(a) Visit Roboflow:

- Go to Roboflow Universe.
- Browse or search for the dataset you need.
- Download the dataset in YOLO format.

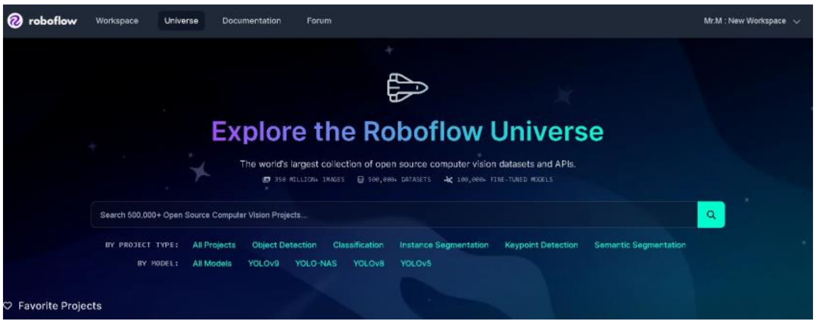


Figure 3.33 Roboflow

2. Labeling the Dataset

(a) Label the Images:

- Go to MakeSense.AI.
- Upload the images you downloaded from Roboflow.
- Manually draw bounding boxes around the objects you want to detect. Ensure you label at least 20 images for a minimum dataset, but more images will result in better training accuracy.
- Download the labeled dataset in YOLO format.

3. Organizing the Dataset

(a) Create Directory Structure:

- Create a main directory for your dataset.
- Inside this main directory, create two subdirectories: images and labels.

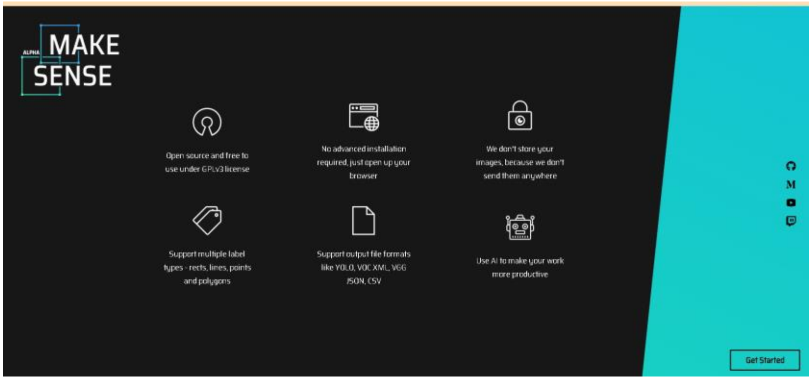


Figure 3.34 Makesense.ai

- Inside both the images and labels directories, create two more subdirectories: train and val.

(b) Organize Images and Labels:

- Move the training images to images/train and their corresponding label files to labels/train.
- Move the validation images to images/val and their corresponding label files to labels/val.

4. Training the Model

You can train the model using either Python on your local machine or Google Colab. Below are the steps for both methods.

Training Using Python

(a) Install Required Libraries:

- Ensure you have Python installed on your machine.
- Install the necessary libraries as shown in figure 3.34.

(b) Prepare the Training Script:

- Create a Python script (e.g., train..yolo.py) with the following content as shown in figure 3.35.

```sh
pip install ultralytics opencv-python-headless
```

Figure 3.35 Install the necessary libraries

```python
from ultralytics import YOLO

# Load the model
model = YOLO('yolov8n.yaml')  # Specify the model configuration file

# Train the model
model.train(
    data='path/to/your/dataset.yaml',  # Path to dataset configuration file
    epochs=50,  # Number of training epochs
    imgsz=640  # Image size for training
)
```

Figure 3.36 Training Script

(c) Create Dataset Configuration File:

- Create a dataset.yaml file in your dataset directory with the following content as shown in figure 3.36.

(d) Run the Training Script:

- Run the training script using the command as shown in figure 3.37.

Training Using Google Colab

(a) Open Google Colab:

- Go to Google Colab.

```yaml
train: path/to/your/dataset/images/train
val: path/to/your/dataset/images/val
nc: number_of_classes  # Number of classes in your dataset
names: ['class1', 'class2', ...]  # List of class names
```

Figure 3.37 Dataset Configuration File

```sh
python train_yolo.py
```

Figure 3.38 Run the training script

Figure 3.39 Googlecolab on yolov8.com

(b) Set Up Google Colab Environment:

- Create a new notebook and start by installing the necessary libraries as shown in figure 3.39.

(c) Upload Your Dataset:

- Upload your dataset to Google Colab. You can use the following code to unzip your dataset if it is in a zip file as shown in figure 3.39.

(d) Prepare the Training Script:

-

(e) Download file coco128.yaml:]

- Download file coco128.yaml for configuration as shown in figure 3.37.
- Change the class number, class name based and the path of the files based on your case as shown in figure 3.42.
- Rename the file and added to yolov8/data as shown in figure 3.43.

```python
!pip install ultralytics opencv-python-headless
```

Figure 3.40 Python installation

Figure 3.41 Unzip file if needed

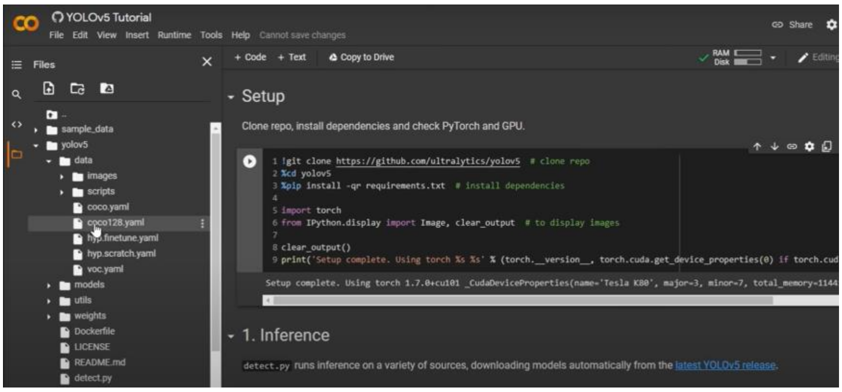


Figure 3.42 Download file coco128.yaml

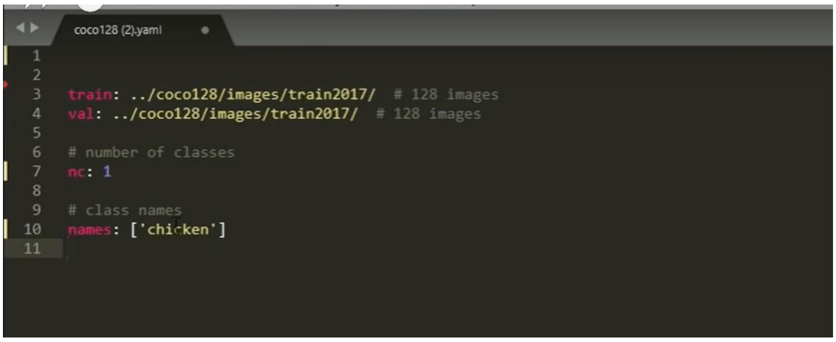


Figure 3.43 Change the class number and class name

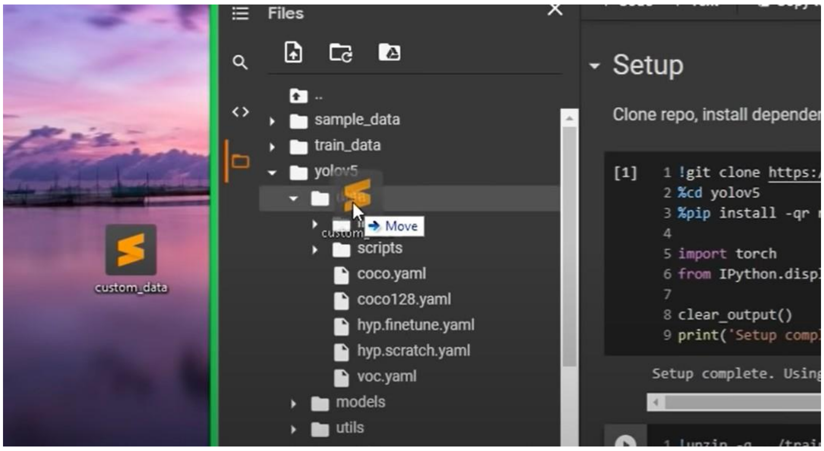
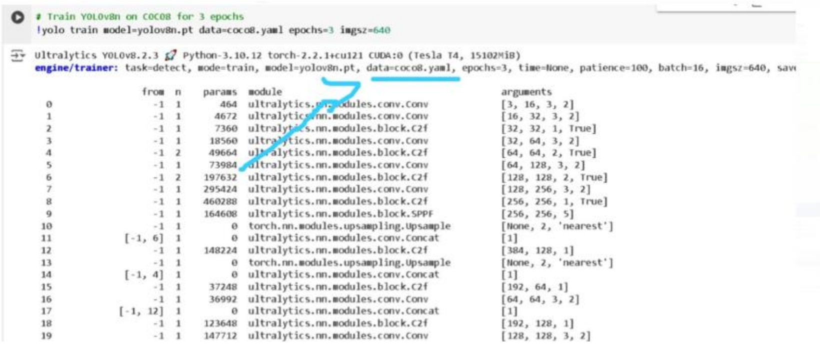


Figure 3.44 Adding renamed file for class number



```
# Train YOLOv8n on COCO8 for 3 epochs
!yolo train model=yolov8n.pt data=coco8.yaml epochs=3 imgsz=640

Ultralytics YOLOv8.2.3 🚀 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
engine/trainer: task=detect, mode=train, model=yolov8n.pt, data=coco8.yaml, epochs=3, time=None, patience=100, batch=16, imgsz=640, sav

                   from  n    params  module                                       arguments
  0                  -1  1       464  ultralytics.nn.modules.conv.Conv             [3, 16, 3, 2]
  1                  -1  1      4672  ultralytics.nn.modules.conv.Conv             [16, 32, 3, 2]
  2                  -1  1      7360  ultralytics.nn.modules.block.C2f             [32, 32, 1, True]
  3                  -1  1     18560  ultralytics.nn.modules.conv.Conv             [32, 64, 3, 2]
  4                  -1  2     49664  ultralytics.nn.modules.block.C2f             [64, 64, 2, True]
  5                  -1  1     73984  ultralytics.nn.modules.conv.Conv             [64, 128, 3, 2]
  6                  -1  2    197632  ultralytics.nn.modules.block.C2f             [128, 128, 2, True]
  7                  -1  1    295424  ultralytics.nn.modules.conv.Conv             [128, 256, 3, 2]
  8                  -1  1    460288  ultralytics.nn.modules.block.C2f             [256, 256, 1, True]
  9                  -1  1    164608  ultralytics.nn.modules.block.SPPF            [256, 256, 5]
 10                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 11             [-1, 6]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 12                  -1  1    148224  ultralytics.nn.modules.block.C2f             [384, 128, 1]
 13                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 14             [-1, 4]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 15                  -1  1     37248  ultralytics.nn.modules.block.C2f             [192, 64, 1]
 16                  -1  1     36992  ultralytics.nn.modules.conv.Conv             [64, 64, 3, 2]
 17            [-1, 12]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 18                  -1  1    123648  ultralytics.nn.modules.block.C2f             [192, 128, 1]
 19                  -1  1    147712  ultralytics.nn.modules.conv.Conv             [128, 128, 3, 2]
```

Figure 3.45 Training Script

(f) Run the Training Script:

- Run the script after changing coco128.yaml to the renamed file used as shown in figure 3.44.

By following these steps, you will be able to train a custom dataset for object detection using YOLOv8, whether you choose to do so on your local machine with Python or using Google Collab for a cloud-based solution.

3.2.7 Troubleshooting

When training a custom dataset using YOLOv8, several problems can occur that need to be resolved. One of the most frequently occurring issues is that the required libraries might not behave properly. This might be because of several issues like wrongly installing it, the problem of version number, or improper libraries. To resolve this, you should first ensure that you have the following libraries installed correctly on your machine. If any problem exists, in case you have installed the problematic library via pip then remove it by typing the command pip uninstall ¡library name¿ and again install it using pip install ¡library name¿. It is also advisable to search for the updates of the library that can solve the bugs or such type of compatibility problems. Other things that can be done include updating the Python environment and or any other dependency to the latest version.

Another concern that can come up is associated with the values of images in your dataset. It is also possible and easy to add wrong images or mislabeled images, in such a situation, the training process becomes ineffective or provides poor outcomes. To fix this, one must go through the dataset and make triple check all images and correctly arrange them in train and val directories. Some examples of the nowadays tools applying are MakeSense. AI to visually inspect and correct the bounding boxes In this regard, the following steps were performed. Also, make sure the structure and content of your dataset configuration file, usually, 'dataset. yaml', matches your dataset. Should the errors persist once they have been defined, it may be helpful to start with a smaller subset of your data to identify and correct any specific issues.

File path errors are also common, particularly if the paths specified in your scripts or configuration files are incorrect. Double-check that all file paths are accurate and that the files are accessible from your working directory. If you encounter a ”file not found” error, ensure that the paths in your dataset configuration file match the actual locations of your training and validation images. Absolute paths can sometimes resolve issues related to relative path errors.

Another factor that may come in the way of a gaming PC is the incompatibility of some of the various parts of your setup. For example, if you are using old version of a library, or some incompatible hardware part, the training process could fail. To solve this, make sure that all facets such as the YOLOv8 model, OpenCV, or any other necessities are compatible with each other. Consult the documentation of each tool to know about version compatibility of their tools.

If the training process ceases to use the GPU or takes a much longer time than it should, then this might be attributed to some configuration. Make sure that, the version of CUDA and cuDNN needed for your system are rightly installed and properly set up. In Python, the torch has the availability of the specific GPU and provides confirmation of the same. cuda. The available backends depend on the type of GD and include is..available() qualified as the PyTorch backend.

Finally, it often so happens that on training of models, some unknown pre-programmed errors occur or the script of training models is interrupted without any apparent reason; in such cases, the log files can give information as to what went wrong. The user or application logs normally provide rich contextual information of what went wrong, which, in turn, points to the cause of the problem. Exploring the documentation of YOLOv8 and the libraries used in it as well as forums can also help because it is possible to find solutions to some of the problems faced by others.

By systematically discussing regarding these possible issues, one can easily identify and fix issues that might occur during the training of the custom dataset using YOLOv8.

3.3 Smart track

Using the design method introduced throughout mechanical engineering undergraduate studies, a set of systematic steps were used to finish the project. The steps covers from concept screening up to the evaluation to obtain the final concept. Then, the final selection is further improved. The proposed method are presented in the form of flowchart below as shown in figure 3.45.

3.3.1 Product Design Specification (PDS) for Track

Performance

- Primary Function: This way the track has to prove as an ideal ground where the autonomous car and the AI camera can be put through different challenges in the best manner possible.
- Load Capatrack: The constructed track is intended for the miniature car of 1/10 scale to simulate the behavior of a real car on Malaysian roads.

Target Product Cost

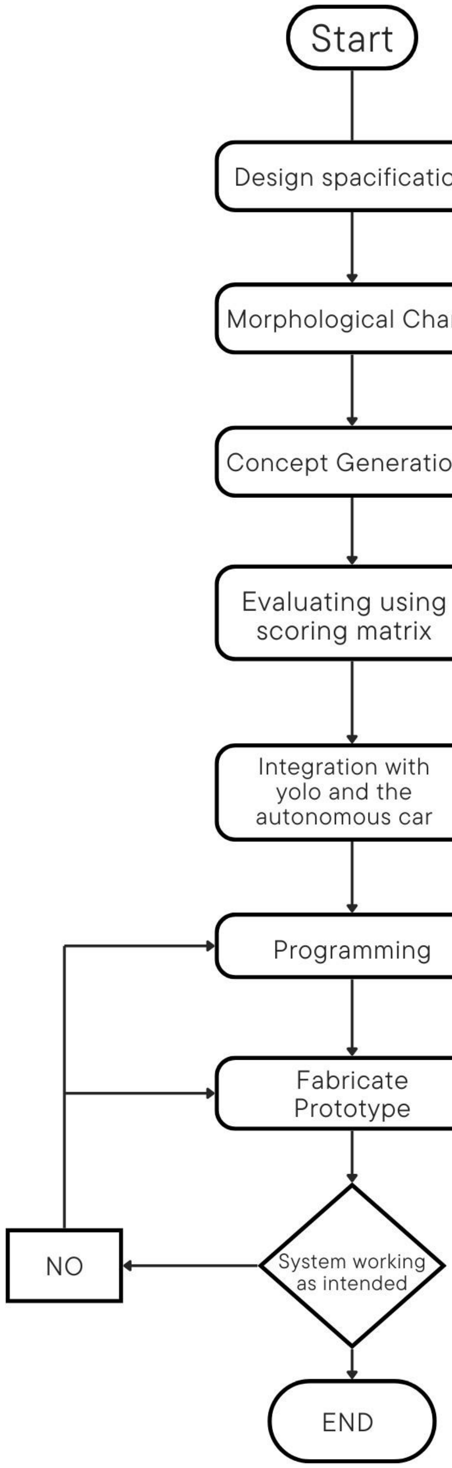


Figure 3.46 Flow chart

- Material Cost: It is thereby asserted that the track is made from plywood for reasons of cost, workability, and indefinite serviceability.
- Manufacturing Costs: Manufacturing is a process that encompasses things like acquiring the right material, slicing it, joining and processing it. Overhead costs are reasonable because the production layouts are simple.
- Budget Considerations: Single thickness plywood and utilizing paint for the markings are inexpensive and permit extensive ascend in versatility for future runs of the track.

Size

- Scale: All the objectives of the track design were met with the utilization of a 1/10 scale of Malaysian road dimensions. Width: The distance between the rails is 62mm, which is equal to the width of the Malaysian two way dual carriageway road at full scale being 6.2 meters.
- Design Flexibility: Follow-on segments are easily assembled and disassembled so that various track patterns can be generated by changing the arrangements of the various segments.

Weight

- Allowable Weight: The cart’s weight is determined by thickness and type of plywood regardless the size in order to allow it to be moved easily without compromising on support.
- Special Materials: The durability required is met by using high-quality of plywood. This material forms a smooth layer for the car testing, tolerates painting for demarcation lines and other attractive colors.

3.3.2 Design Specification (DS)

Planning and detailing in specifying a smart track is one of the complex activities that needs to be undertaken in order to satisfy the inhabitants and to work effectively. Thus the area of the proposed smart track will be around 10 sq km of area, a size considered optimum for variety of facilities/services, yet not too large to be unmanageable. The track is built for a population not exceeding one hundred thousand people at the same time achieving the spirit of the metropolitan and comfort of living. It will be divided into area for living, working, business, education, leisure, and so on, green areas. Housing and residential purposes will consume about 40 percent of the entire space of the metropolis, including multifaceted tenements and individual homes. Social facilities include commercial sections of the track encompassing 20 percent of the space for existence of shopping malls, business offices and entertainment places. Slightly over 10 percent of the land will be dedicated to industrial usage mainly comprising of light industries as well as IT technology parks At the same time, land dedicated to education and recreation will encompass 15 percent that includes schools, universities, parks and playing fields. The rest 15 percent will be covered with greenery providing sufficient open space for leisure and adequate amount of green coverage conforming to the requirements of environmental concern.

Transportation system will in the smart track will comprise of proper layout and design with strategic main roads , access roads, regional and minor road network. The proposed transit system will have predictions, bus stops, trams, and an underground metro system to improve public transport experience in Prague. Autonomous vehicle lanes will also be accompanied by traffic management and safety with the help of our YOLOv8 approach. Utilities will AC accessories the smart water management system to track water usage and leaks, ensuring sufficient distribution. In the case of electritrack, it will pull from a smart grid that will include the use of renewable power resources like the solar and wind energy with smart meters to aid in energy management and tracking. Self-driving trash trucks and sorting plants will apply AI to determine optimal routes for garbage transporting and recycling. Free WiFi and 5G coverage will be provided in the track to provide connectively to IoT devices and to allow continuous interaction.

Smartness will be in its application such as smart homes with IoT enabled for energy and security as well as home automation. The surveillance cameras and drones to remain operative will be AI-driven with collected data being stored and processed subjecting to rigorous access restrictions for security purposes to enhance public safety. Effective progress in environmental monitoring shall involve installation of sensors in the track that will be used in monitoring air quality, noise level, and weather conditions and a central system for evaluation of the results and subsequent action. The healthcare sector will be integrated with intelligent digital healthcare solutions such as tele-health, diagnosis and patient care tracking for efficient and effective healthcare.

Environmental aspects are sustainability-related and strive for carbon neutrality through green building types, large green zones, as well as electric cars. Buildings will utilise energy-efficient materials and technologies, from smart lighting to climate control systems. Rainwater Systems Grey water recycling also the fundamental system help to solve water conservancy. This will include tight cybersecurity protocols like firewalls and encryption to ward off data leakage, running regular checkups for vulnerabilities - making both system-wide works without a flaw. The track must also have early warning systems for natural disasters and well-planned evacuation routes to accommodate various emergency response systems.

As for the aspect of the community and living, smart track will have advanced specialized IC learning establishment and research centre. Recreation: It thus became evident that parks, sporting facilities, and cultural facilities enhance the quality of living of people because they promote exercise. To meet this goal, all these specifications are integrated into the smart track that will help people to have a comfortable, quality living standard enhanced with efficiency displayed through resource saving, which is made possible through new technologies. Another indispensable factor that should not go unnoticed when explaining this concept is the construction of this project to a one-tenth scale for realistic exposure. This has been down-scaled to ensure adequate usage if the planned technologies in the integrated autonomous vehicles under the current test environment before setting a perfect test environment for the systems to perform at optimal capabilities. The proposed actual physically integrated smart track model will be 1/10 of the actual physically integrated smart track and it will incorporate all the complexities of the real integrated smart track,

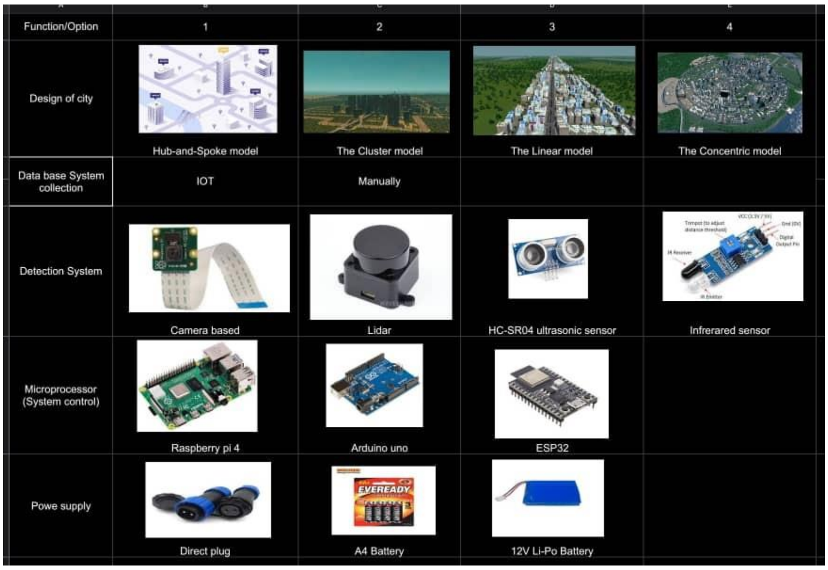


Figure 3.47 Morphological Chart

hence it will be possible to carry out actual testing of AVs and all other advanced smart technologies as it will closely mimic the real smart track.

3.3.3 Morphological Chart

Figure 3.46 is the morphological chart with all the possible solutions for each function

3.3.4 Concepts

Concept 1: Hub-and-Spoke Model Track having Lidar, Raspberry Pi Processor, and 12V Li Battery The given model track design is based on Hub-and-Spoke structure, Lidar technology is used for better mapping and obstacle recognition and Raspberry Pi processor for computational and controlling purposes. This concept avails cost-effectiveness because Raspberry Pi and Lidar are cheaper than the high-end systems. Nevertheless, using a 12V Li battery as the source of power has some challenges like low time of operation and a chance of frequent recharging which hinders the extended smooth operation without being charged. Also, Lidar offer precise distance measurements; however, it may be affected by some factors such as rain or fog.

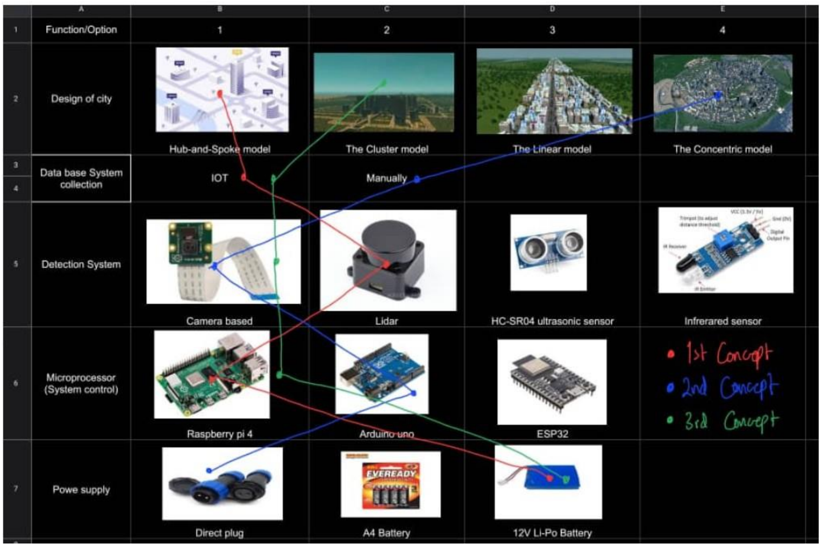


Figure 3.48 Morphological Chart Concept Selection

Figure 3.49 Concept 1

Figure 3.50 Concept 2

Concept 2: Naturally, there is the Concentric Model Track that comes with a Camera-Based System and the Power Supply for the Arduino Uno. The design pattern of Concentric model track is based on the camera perception system for visual recognition and associated with Arduino Uno as the processing unit and power source. This concept is easy to address in relation to simplitrack and implementable in light of Arduino which is plug and play as well as uses low power. However, Arduino uno's use might restrict the ability to process images or any other multivariate data to full processors such as Raspberry Pi. In addition, the camera-based systems depend on the lighting conditions and the resolution of the cameras that are involved making it possible to have reduced detection rates especially under poor lighting conditions or use of low resolution cameras.

Concept 3: Cluster Model Track with Camera-Based System, Raspberry Pi Processor, and 12V Li Battery The Cluster model track design integrates a camera-based system with a Raspberry Pi processor and a 12V Li battery for power supply. This concept combines the computational power of Raspberry Pi with the flexibility of a camera-based perception system. Raspberry Pi's capabilities support complex image processing algorithms, enhancing object detection and navigation tasks. However, similar to Concept 1, the use of a 12V Li battery introduces challenges related to battery life and recharging intervals, impacting the autonomy and operational efficiency of the system. Moreover, while Raspberry Pi offers greater processing power

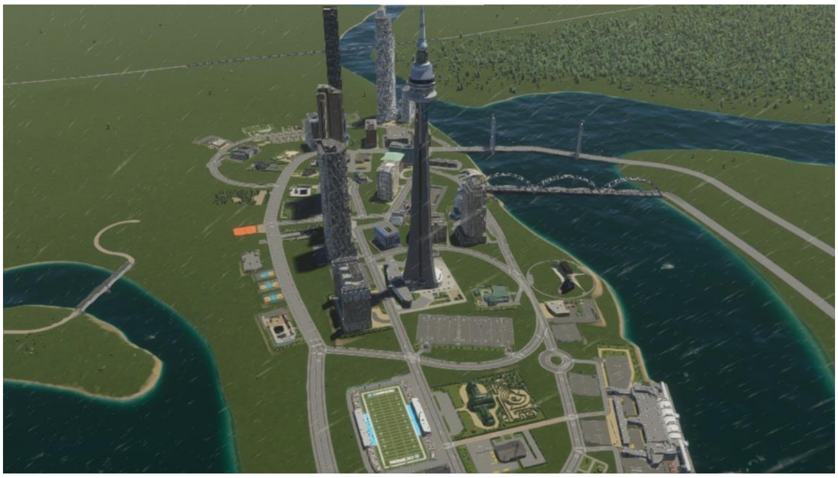


Figure 3.51 Concept 3

than Arduino Uno, it also requires careful power management to optimize performance and battery life.

3.3.5 Concept evaluation

Based on the concept evaluation using weighting factors and rating factors, Concept 3 emerges as the most favorable option with a score of 7.6 out of 10. This score reflects its alignment with the specified criteria and relative importance assigned to each factor. Concept 3, which involves a Cluster model track utilizing a camera-based system, Raspberry Pi processor, and 12V Li battery, scored highest due to its balanced advantages in computational power, flexibility in perception capabilities, and potential for scalable deployment in smart track environments. This evaluation underscores Concept 3 as the preferred choice for further development and refinement towards implementing autonomous vehicle solutions within a smart track framework.

Criteria	Weighting Factor (W.F.)	Concepts					
		1		2		3	
		Score	Score × Weightage	Score	Score × Weightage	Score	Score × Weightage
Ease of Fabrication	0.05	8	0.4	8	0.4	8	0.4
Cost	0.05	7	0.35	9	0.45	8	0.4
Function Efficiency	0.2	4	0.8	7	1.4	8	1.6
Life span	0.1	6	0.6	5	0.5	7	0.7
Security	0.2	7	1.4	2	0.4	8	1.6
Ease of Maintenance	0.1	7	0.7	8	0.8	6	0.6
Power Demand	0.1	2	0.2	6	0.6	5	0.5
Complexity	0.05	4	0.2	5	0.25	6	0.3
Remote Monitoring	0.1	6	0.6	4	0.4	7	0.7
Speed	0.1	3	0.3	3	0.3	8	0.8
Total	1		5.55		5.5		7.6

Figure 3.52 Concept evaluation

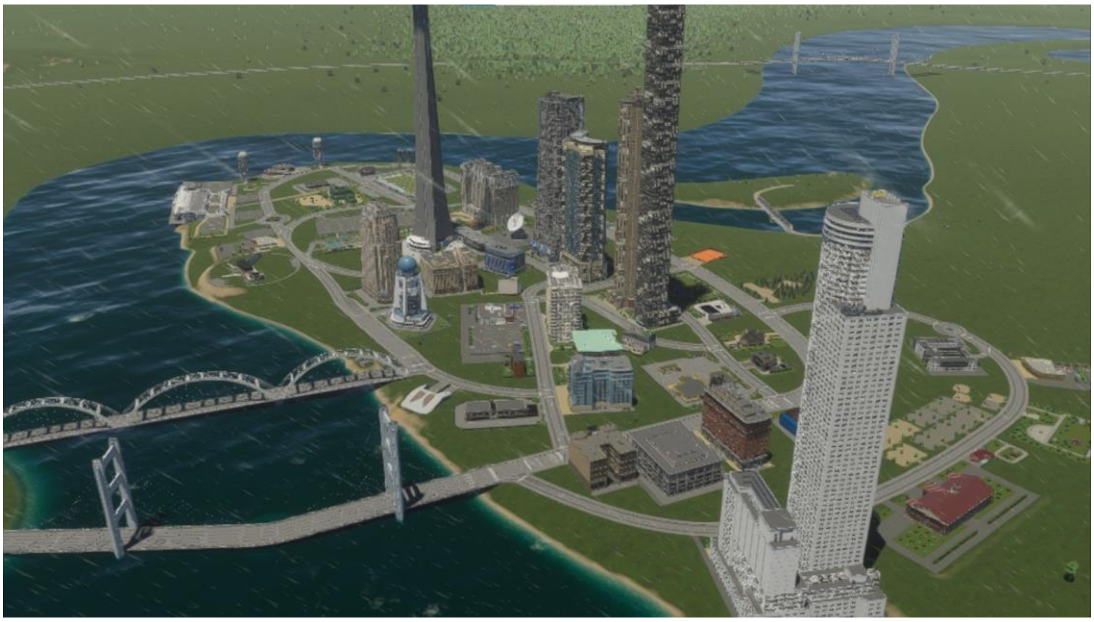


Figure 3.53 Concept Selected

Figure 3.54 Concept Selected Top View

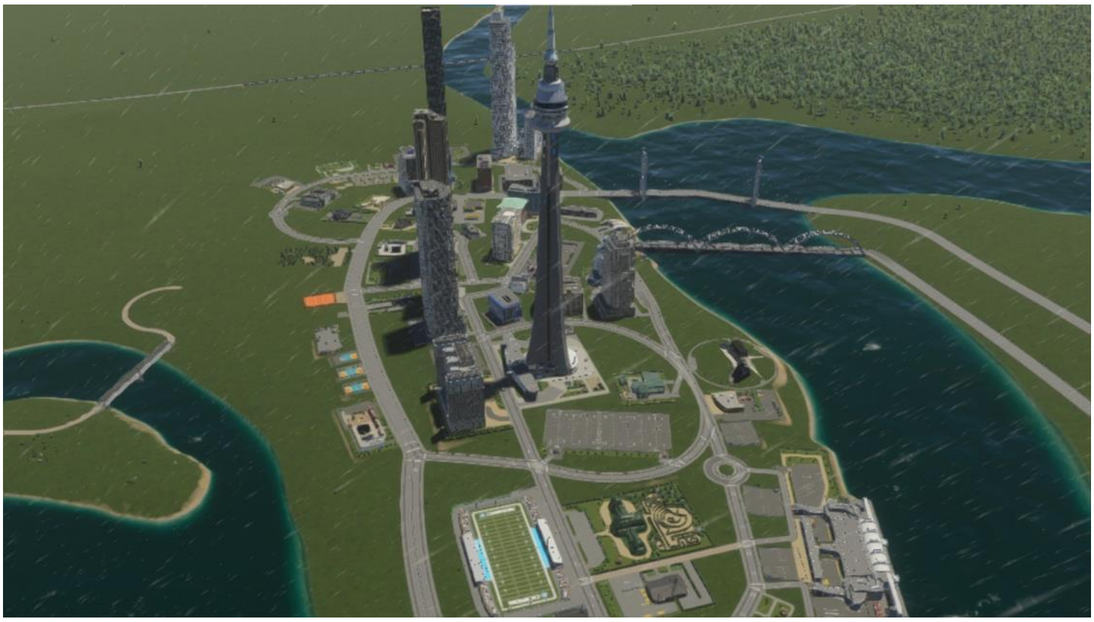


Figure 3.55 Concept Selected Side View

3.4 Design 1/10 scale track for KTP (Knowledge Transfer Program

3.4.1 Morphological chart

Figure 3.54 is the morphological chart for the KTP (Knowledge Transfer Program) which is made to be modular and customisable for the Kids or other people who may be interested.

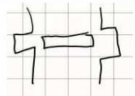
Functions/Options	1	2	3	4
Material	Plywood	PVC Foam Board	Acrylic Sheet	Medium Density Fiber board
Connection			Glue / Screw	
Road Planning	Carpet	Sticker	Paint	Plywood Sheet
Electronics	Full City	Simple City	Smart City	Single District
Power	AA Battery	Outlet	Solar	Power bank

Figure 3.56 Morphological chart for KTP

3.4.2 Concept

Thus, the general idea for the KTP (Knowledge Transfer Program) track design is to prepare an environment that would resemble schools where teachers illustrate the notion of autonomous vehicles to kids. The track should be uncomplicated but engaging to enable them take their attention from the games and learn what is important. But one of the easiest and most practical patterns is an oval or a figure-eight configuration. These shapes are articulated not to involve complicated junctions, and offer sustained movement; therefore appropriate for a race track configuration.

To elaborate on material, thus the choice of medium density fiber board- MDF or plywood since they are strong; can be easily shaped and painted to the base of the track. To provide lanes and boundaries, commonly, tape or paint on the floor can sufficiently mark the lanes. To increase the level of realism some lighting for signals and street lamps should be included also increases the facility’s attractiveness

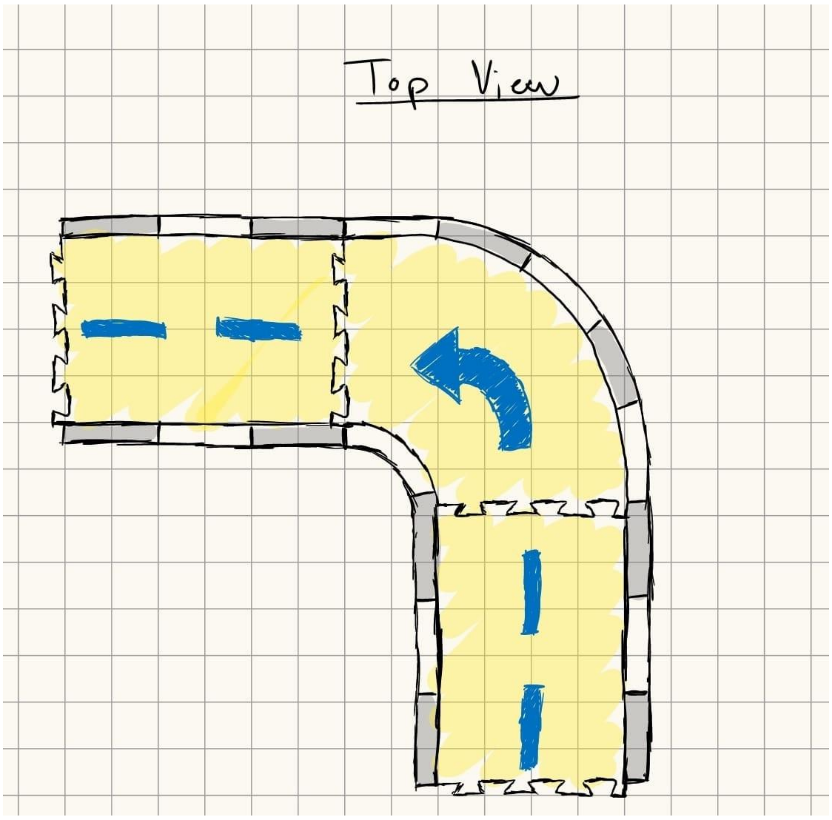


Figure 3.57 Concept of the track

and contributes to the real road conditions imitation. Objects like miniature buildings and structures or seating areas, small audience stands, and miniature chequered flags should be placed all over the track in order to create density of the environment. Pre-made ones which could be 3D printed or bought can also save a lot of time and also give a more professional look if incorporated with little additional effort.

That is why for the surrounding environment, as well as small trees, signboards, and a pit stop area will be useful in making the environment as realistic as possible for the kids. Also, installing the sensors along the track for speed measurement, or laps counting purposes, can also be a part of the educational procedure as how coding and sensors work in self-driving cars for children. The overall goal from this point of view is to ensure that the activities that the students are engaged in are all touch and visually related to ensure that they demonstrate the essence of the self driving cars and coding.

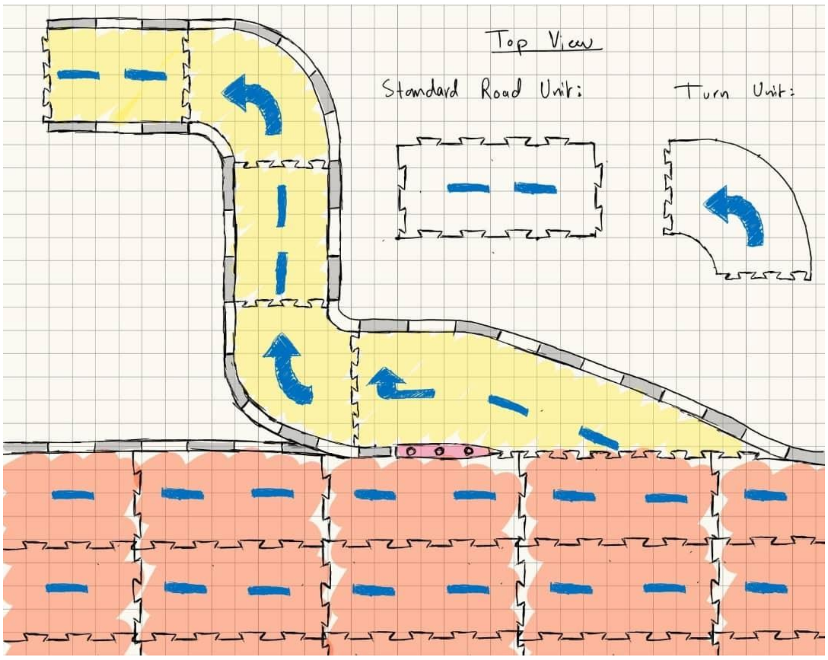


Figure 3.58 Concept of the track in details

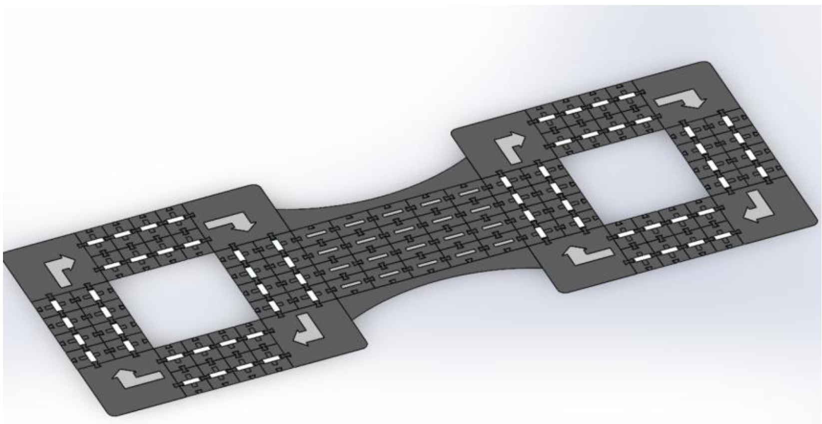


Figure 3.59 3D Model of the track

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Yolov8 using laptop camera

The first stage of our work was to perform an experiment with the YOLOv8 object detection model using the laptop camera. This was done due to convenience because a laptop could easily be carried around with the added bonus that it has relatively high processing power that makes it perfect to test the model in. This evaluation proved to be quite positive and offered the possibility to understand the effectiveness of YOLOv8 under real-world conditions in the first tests.

Fast Response Time Probably the most significant outcome was the deviation of the YOLOv8 model from being slow to incredibly fast. Because of the high processing power of the laptop processor, the object detection was fast and did not consume a lot of time. The laptop offered enough computing power due to a multi-core processor and the sufficient amount of RAM, and it was capable to process the YOLOv8 model without noticeable lagging. This is especially important in applications like the self-driving vehicles and surveillance where the response time should be fast so that the newly arriving frames can be inspected and the objects detected on time.

High-Resolution Camera The high-resolution camera also proved helpful in the efficiency of the YOLOv8 model. The fact is that a camera records video with smooth and clear motion, which means that the model would have high quality input data. These aforesaid features are vital in view of the fact that high-resolution imagery allows for better object identification together with them being localized precisely within the frame, particularly where there are more than a few objects in the frame or their presence is somewhat obscured. Yolov8 was able to identify smaller items and perform better than the previous tests due to the quality of the high-res camera.

Accuracy and Precision In the tests, YOLOv8 model successfully proved to be accurate and precise in terms of identifying the objects on the screen. In terms of correctness of identification and classification of objects within the camera's field of view it was as follows: The bounding boxes created around the detected objects were accurate. It is important for autonomous systems where missing an object or identifying it wrongly can be disastrous to a particular system.

Real-World Implications The above tests of the laptop camera with YOLOv8 signal great potential in applying the algorithm in different real-world use cases. For example, under the framework of a smart track, this kind of feature of the model, namely, the real-time and high-precision identification of objects, can improve many functions in the track, such as traffic control and monitoring of public security. For real time monitoring and decision-making related with customers and the organization, the fast response time, high accuracy is very useful.

Smart Auto Mobile/Smart Urban. The positive results also signify that with more enhancement and incorporation with other devices, such as depth cameras and LIDAR systems, YOLOv8 can be one of the fundamental components of sophisticated, self-sufficient systems. In the context of the autonomous car project, that entails using YOLOv8's functionalities to make the vehicle to move safely and optimally in the smart track context. Yes the car can integrate the ability to process real time data from the cars' sensors and make various real time decisions on the fly such as speed, steering and avoiding of objects which makes this type to be more safe and reliable as compared to type one.

In the case of integration of YOLOv8 with a number of important and essential smart infrastructural components of the smart track, a marked improvement in the smart track function will be observed. For instance, the intelligent traffic control technology that can help to minimize traffic jams and increase the general security level on the roads by using the real-time control of traffic intensity. Likewise, increasing surveillance also helps in the prevention of insecurity as it provides the security organs with tools to counter threats as soon as possible.

Figure 4.1 Yolov8 using laptop

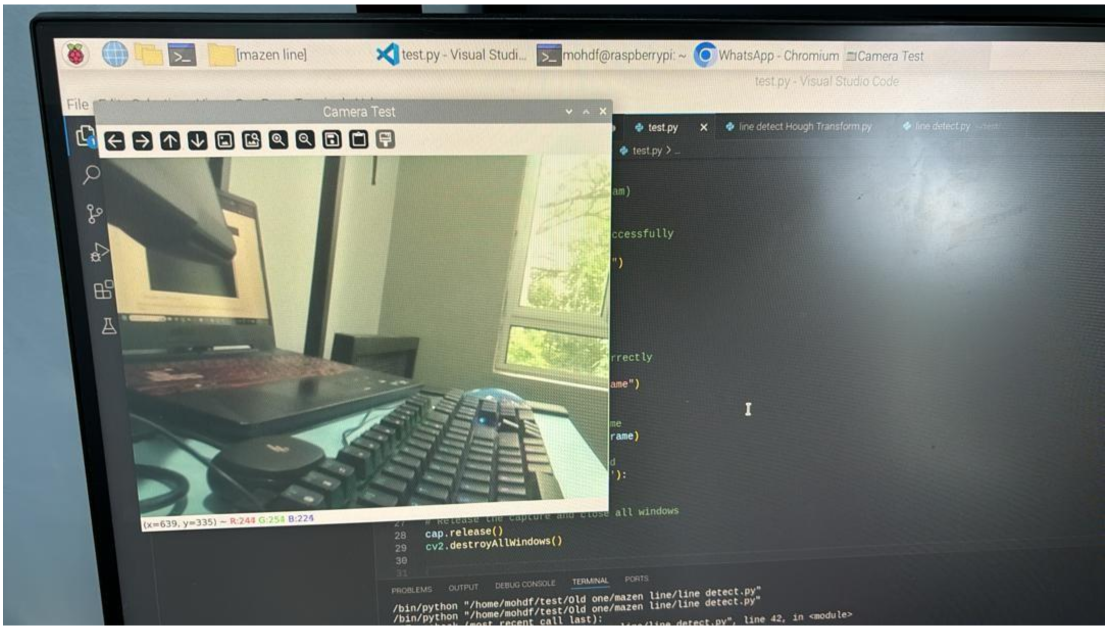


Figure 4.2 Raspberry pi camera configuration

4.2 Code for both autonomous car and track

4.2.1 The response time

Hence, the type of hardware used in the implementation of line detection for the autonomous vehicle highly influences the system, for instance, the response time. The response time is quite essential in the AV approach due to its apparent relationship with the car's capatrack to respond to alterations within its environment so as to minimize potential risks in maneuvering.

Laptop Performance Overall, results of using a laptop with a high-resolution integrated camera as well as a highly processed graphics card are good for the line detection tasks. With the high-end specifications on the laptop, simple image processing can be accomplished in real time to detect lane lines followed by the necessary reaction from the system. A fast processor can execute the image processing algorithms in a very efficient manner, thus, leading to a small amount of latency. This fast response time is particularly useful in applications where the self-driving car has to decide on the identified lane markings instantly.

For instance, the line detection system obtained on the laptop could manage the frames at 30-60 fps provided that the configuration and optimization of the code would allow it. This high frame rate also makes it possible for the vehicle to pass on the most recent information concerning the position of the lane lines to the vehicles' navigation systems for a quick and efficient adjustment of the vehicle's steering.

Raspberry Pi Performance However, using the line detection system on a Raspberry Pi also exposes some disadvantages because the Raspberry Pi is less effective than the laptop used in this work. Thus, Raspberry Pi, though flexible and cheap, has fewer processing capabilities compared to the current laptops. This is due to the fact that more time is taken in order to perform image processing tasks as a result of the large number of memory directories that are created by JSP files.

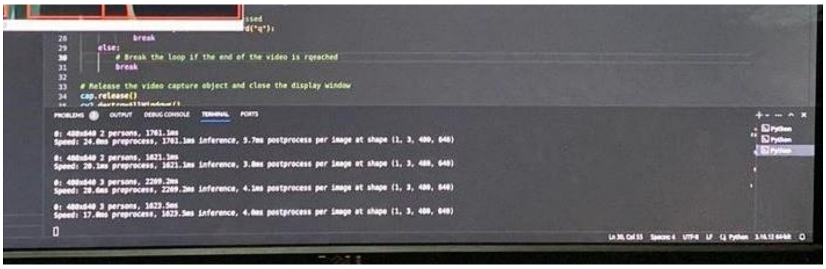


Figure 4.3 The response time for raspberry pi

As for the Raspberry Pi, the same line detection algorithm takes approximately 10-15 fps when evaluating the system’s performance. This lower frame rate results in a delay in detecting and responding to lane lines which is quite apparent.The lower processing power also means that more complex image processing tasks might take longer to complete, which could impact the overall performance of the autonomous vehicle in dynamic environments.

4.2.2 Line detection using open cv

The first result from testing the autonomous car is the capability of detecting, mapping and following the lines using the OpenCV together with Raspberry Pi. The line detection system works because the basic elements are mathematical equations that determine the form of the street and the distances of the sections. This computational approach provides a more robust and accurate interpretation of the layout and thus affords easier understanding and forecast of the car’s behavior in the road.

Using the Hough Line Transform and the edge-detection technique, the system is also able to detect road markings reliably even at night time. The fact that the detection system is mathematically defined enables the car to determine position of the detected line in relation to the car hence the path to be followed. This will make the car sit in the middle of the lane as it would be in real world driving. The performance of the system is excellent when making turn and in areas involving intersection since it determines the curvature of the road in real-time.

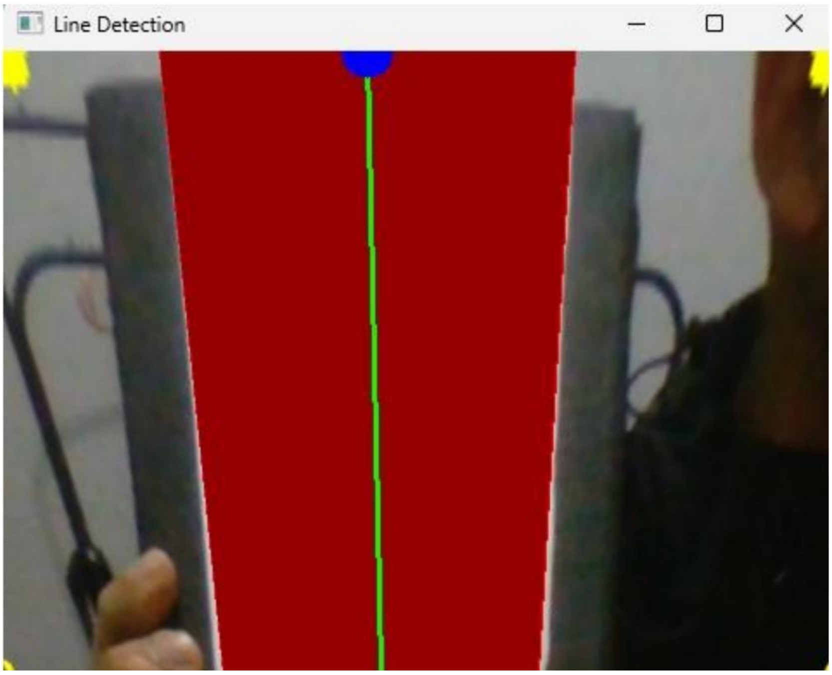


Figure 4.4 Line Detection

Meaningful, this is illustrated by the results as presented in the figure here below where stability of the line detection mechanism is clearly portrayed. The figure also shows how the system detects the lane and measures the distance between the vehicle and the middle of the lane. Such information is processed instantly, which allows the car to quickly change its course rapidly.

In conclusion, the efficiency of the line detection system is highlighted with additional possibilities for further use and development of AUVs identified as shown in the figures.

4.2.3 Traffic light detection

Now for traffic light detection, Raspberry Pi was employed to detect and analyze the traffic signals through an AI camera in real time. Using sophisticated algorithms for the detection of objects on a frame, the system is able to identify the states of traffic signals effectively as red, yellow and green. The AI camera need to be able to recognize traffic lights from different angles and lighting, this is made possible

Figure 4.5 Line Detection Final Results

by training a machine learning model like YOLO to the car's camera so that it will be able to recognize traffic light from all these angles.

Traffic light detection system analyses the captured video feed from the camera to determine a state of the traffic signal. When a red light show up, the system issues a stop signal and stops the vehicle at a considerable distance from the intersection. For instance, if the color red is identified at 10m, the system makes the vehicle to come to a stop at 2m in line with real world driving conditions. This precise control is obtained from distance calculations involved in object size and position in the video feed.

The figure below also captures the reliability of the detection accuracy and the response time and the above results show that improved detection accuracy and response times are attainable with the proposed heuristics. The system is able to tell the difference between red, yellow, and green effectively even when there are other objects of the red, yellow or green color in the same area. The response time is less than a second and the control decision-making is on the spot for safe operation in the changing surroundings.

Therefore, the success of traffic light detection points to the possibility of integrating vision systems supported by AI with sound programming. It can be regarded as a part of the way towards creating a self-driving car that can adapt to the existing traffic-control infrastructure as seamlessly as possible.

4.2.4 Road sign detection

Moreover, road signs were detected using Raspberry Pi and AI camera to capture the signs and analyze the information so as to present it to the navigation system of the AV. This capability enables the car in regard to traffic signals and warnings including; speed limit, stop sign and directions among others.

The detection system uses existing ML algorithms like YOLO with the aid of a labeled database of road signs. Being a part of the vehicle, the camera translates the video feed, interprets the shapes, colors and text on the road signs. When a sign is identified, it is sorted into groups, following which it is identified as being a stop, yield or speed limit. This information is sent to the vehicle’s control system for the correct response which may include, stopping, reducing speed, or varying speed.

To increase the detection rate, all necessary signs were marked distinctly and large enough to be recognizable on the 1/10 proportionally scaled road section of the track layout. During the test, the system proved efficient on the track by recognizing different classes of signs of different distance and angles. The detection process is well protected, providing near optimal results in certain scenarios even with changes in lighting conditions.

For instance, as is exemplified in the figure shown below, the response time of the system is within the range of acceptability; actions are started within less than a second from detection. This quick processing leads to an ability for an autonomous car to adapt the current environment promptly to keep safe and efficient.

Road sign detection gives a real-life example of how artificial intelligence as well as computer vision recognizes and optimises the vision system of the self-driving cars. This feature not only increases safety but also gives a user-friendly and

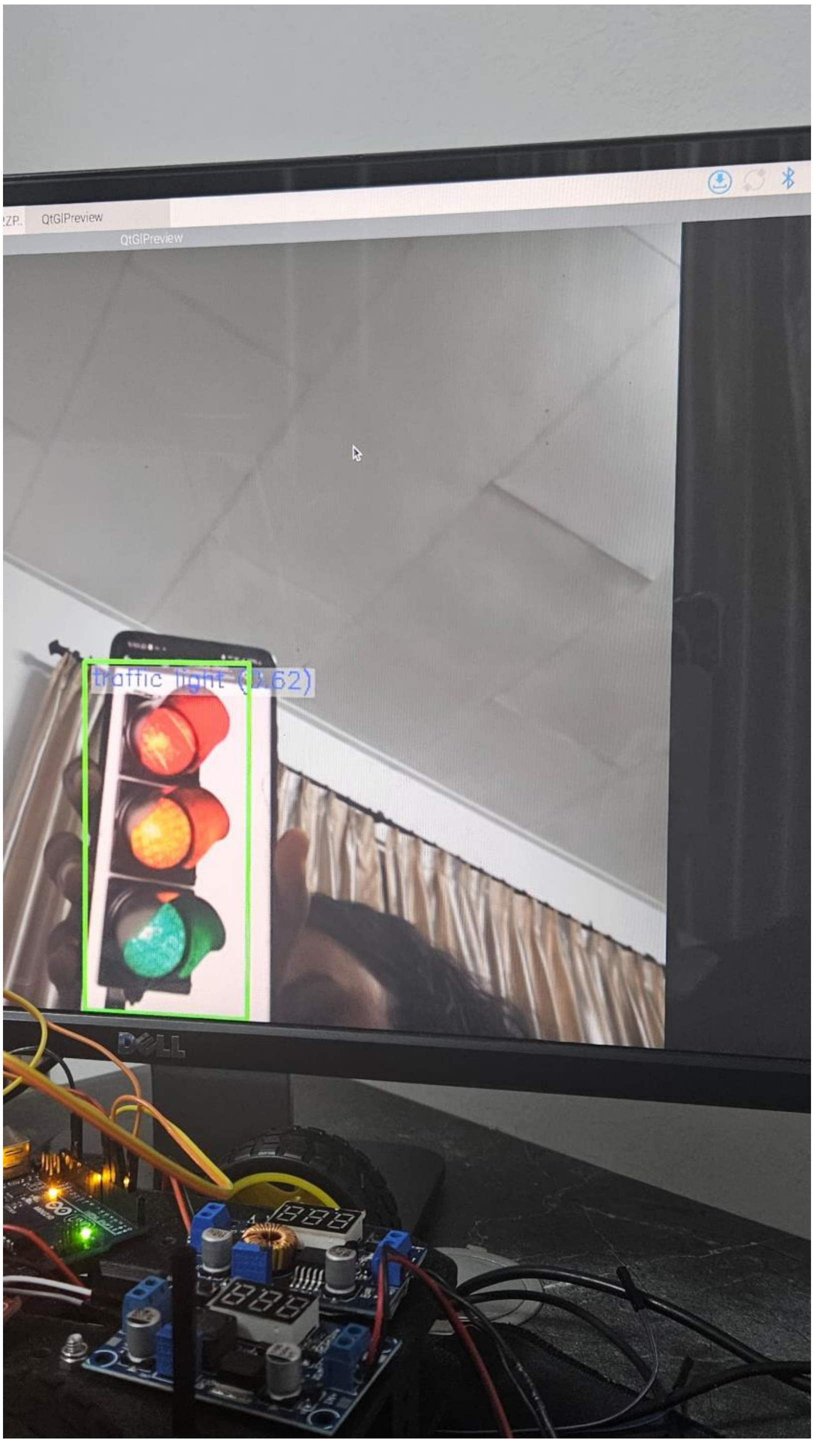


Figure 4.6 Traffic light detection results

informative attraction for participants especially for the KTP that let students figure out the impact of AI in current transport systems.

4.2.5 colors detection

Color detection is a top-level feature incorporated in the self-driving car's decision-making system, allowing the car to discern different colors in order to respond to traffic light signals, the number of lanes, or road signs. This functionality was accomplished by utilizing the Raspberry Pi and the OpenCV libraries which aided in real time color detection.

The system works by receiving the video feed through the AI camera and transforms the video feed RGB to HSV (Hue, Saturation, Value) space because it is less sensitive to the change of lighting than the RGB space. Consequently red, green, blue and yellow color ranges were determined using thresholds and the system begins to detect the objects belonging to these colors.

The colour detecting system also provided efficient results of pattern recognition during the practical trials of traffic signal lights, lanes, and certain road markings and signs. For example:

Traffic Signals: It was evident how the system was always able to differentiate between red, yellow and green light and how this enabled the actions which included halting, decelerating or moving forward if appropriate. Lane Markings: Further, white and yellow lane markings were identified with considerable precision to ensure the car positioning within the racetrack. Road Signs: Differentiated in color like blue and red signs such as Stop or Yield were recognizable from different distances. The figure demonstrating this analysis reveals the system's capatrack to selectively encircle and emphasize detected colors online in real-time. The detection was very sensitive, which took less than one second from the time the software recognized an item until it urged the body into action.

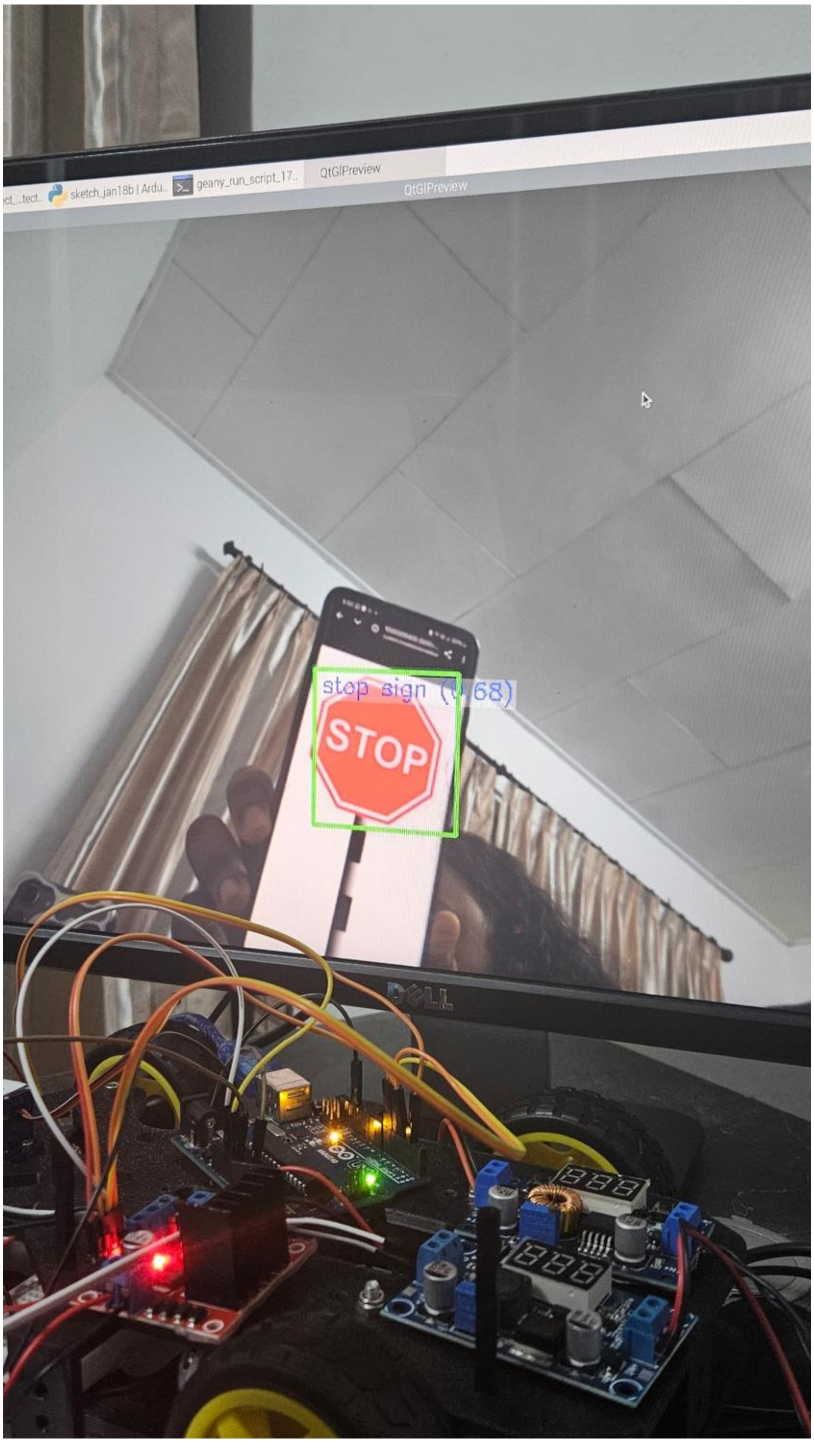


Figure 4.7 Road sign detection results

Challenges and Adjustments On and off during testing, shadows and lighting influenced the detection in certain areas. In order to overcome this, adaptive thresholding approaches and dynamic calibration were added to the system to ensure that device functionality can change with the surrounding environment.

4.2.6 Final assembly and circuit results for the car

The integration of the frame structure and the circuit works as the final step to integrate all subassemblies and subcircuits to ensure stable operation and reliable reactions to real-world conditions. These two main controllers are a Raspberry Pi and an Arduino Uno connected, and powered through a vibrant circuit. Electric supply is from four 3.7V batteries that provide the means needed to supply power to all the parts in the car.

To tackle high voltage differences between the Raspberry Pi and Arduino Uno two voltage dividers are included in the design. These dividers make it possible for Raspberry's 3.3V signals while the Arduino has 5V signals allowing for proper compatibility between the two controllers.

Depending on the design of the car under construction there will be two motors each connected to a motor driver that is responsible for the movement of the car. The motor driver in the circuitry operates from commands from the Arduino Uno while the Raspberry Pi has the input signal processed and forwarded. The motors ensure a perfect power and direction to be produced affording the car to follow lines as well as direct the car through the track. Furthermore, a servo motor is incorporated in the system for fine tuning the steering this improves the cars agility.

An infrared sensor is mounted on the assembly in order to detect the track lines. This sensor feedbacks information to the Raspberry Pi which in turns interprets the data and allocates movement signals to the Arduino Uno. The IR sensor compliments the AI camera for enhanced accuracy in lane keeping as well as track following.

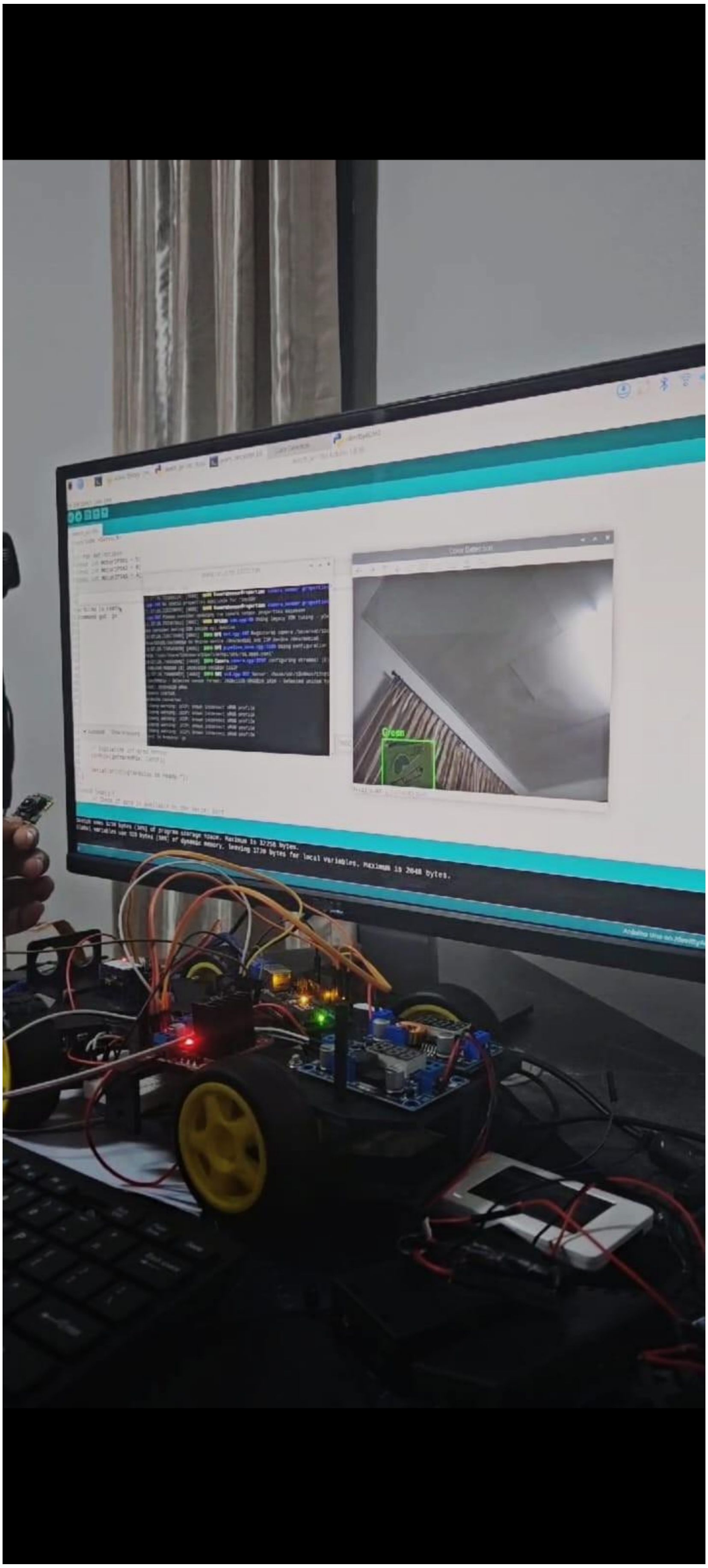


Figure 4.8 Green color detection

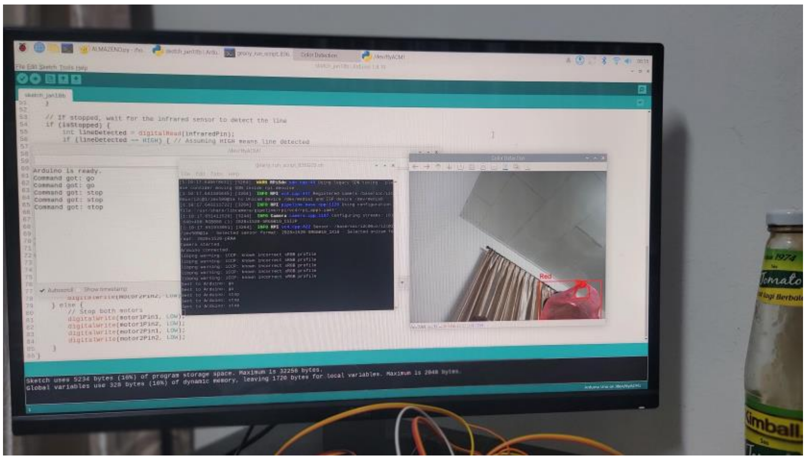


Figure 4.9 Red color detection

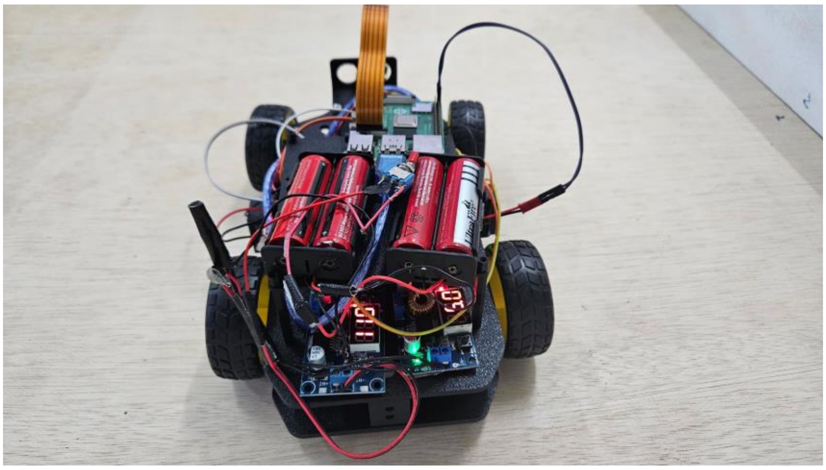


Figure 4.10 Car full circuit rear view

To confirm the circuit operation, the whole circuit was experimented with, and the outcomes exhibited stable working. Voltage dividers were correctly used to maintain communication between the controllers and the motor driver. The infrared sensor was also able to accurately detect lines thus enhancing the performance of the AI camera.

The assembly shows how the mechanical and electronic systems of the self-driving car are built and work hand in hand. By combining these elements, the car showcases a robust system that is both educational and practical, capable of simulating real-world autonomous driving scenarios.

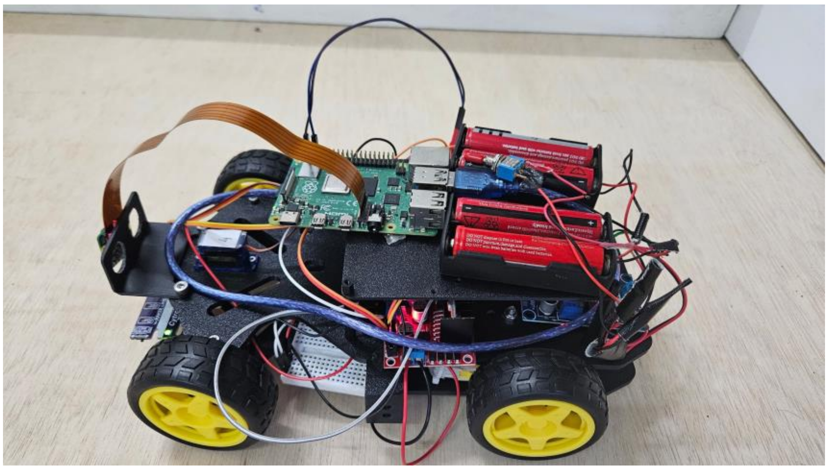


Figure 4.11 Car full circuit side view

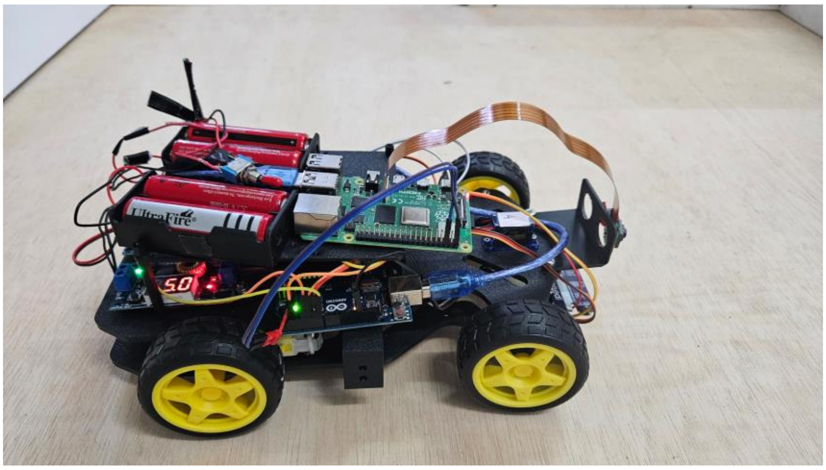


Figure 4.12 Car full circuit other side view

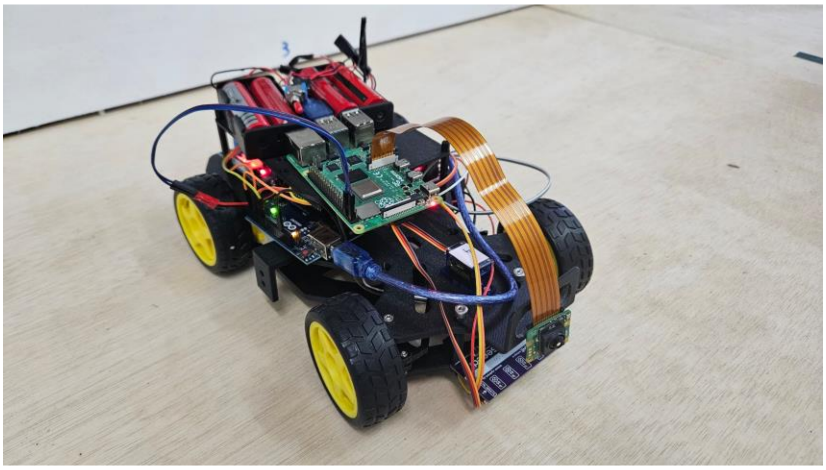


Figure 4.13 Car full circuit angle view

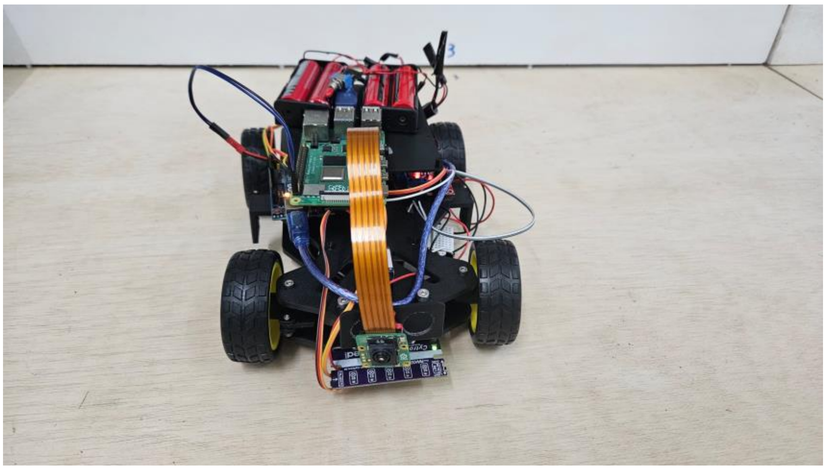


Figure 4.14 Car full circuit front view

4.2.7 Final camera model results

The final camera model combines all first four detection systems: line detection; traffic light recognition; road sign identification; and color detection to create the framework for the operation of the autonomous car. This system uses the Raspberry Pi for image analysis and decision making and issues commands to the Arduino for motor control and operation.

The operations start when the Raspberry Pi AI camera is able to identify some elements in the environment. For instance, when green light is signalled, the Raspberry Pi recognizes this and relays a signal to the Arduino to proceed. It then commands motors to turn on and causes the car to move ahead. In the same way, if there's a detected sign like 'STOP,'; the Raspberry Pi comprehends such a sign and relays the same via a PIN signal 'STOP' that prevents the Car from moving as signaled in the picture below.

Since depth camera was not used, more logic was added to boost the efficiency making it quite accurate. For example, in the vision-based example, if a red traffic light appears in a perceived scene, the car does not automatically responds by stopping. Instead, the Raspberry Pi looks for another condition for example a line of the walkway before issuing a "STOP" command to the Arduino. This helps in ensuring that the car

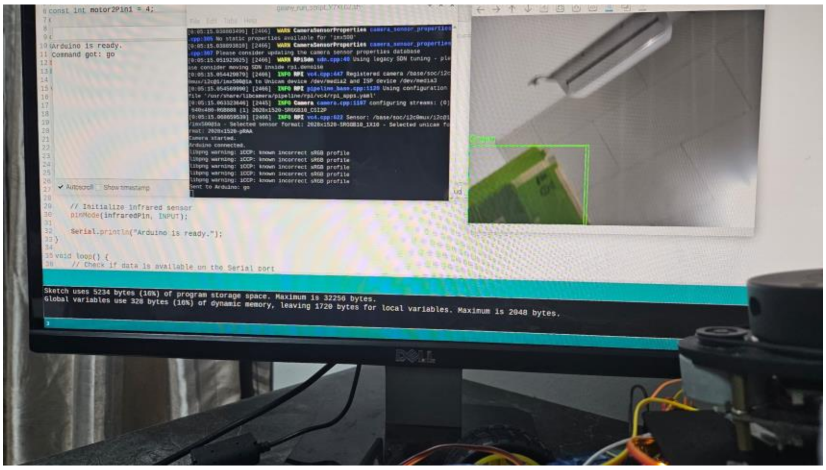


Figure 4.15 Raspberry pi and Arduino results (Green)

stops only at the right place it is programmed to test a real life situation where the cars respond to both signals on the roads and the markings on the same road.

The integrated line detection and road sign recognition are also efficiently performed in the integrated system. This procedure of tracking can be done continuously by the line detection system that help the car to give some sense to only follow the track boundaries that are discoverable. Postprocessing of Road signs is done with a help of pre-trained machine learning algorithms and depending on the detected sign, car slows down, stops or turns. All these functionalities were tried extensively and the system gave reliable results with reference to its identification and response to its surroundings.

This project well demonstrates the use of AI and robotics in the navigation system by integrating a number of detection mechanisms into a single decision making. The results highlight the realism of the system, facilitating interesting and instructional opportunities for students to learn about vehicles. Alongside this project, the car shows how camera based AI, combined with microcontroller systems can be used to enable the car to operate intelligently within a variety of environments.

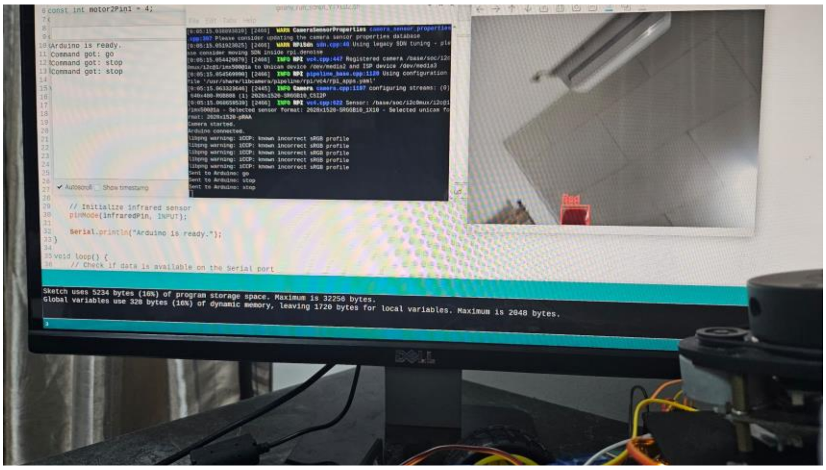


Figure 4.16 Raspberry pi and Arduino results (Red or Stop sign)

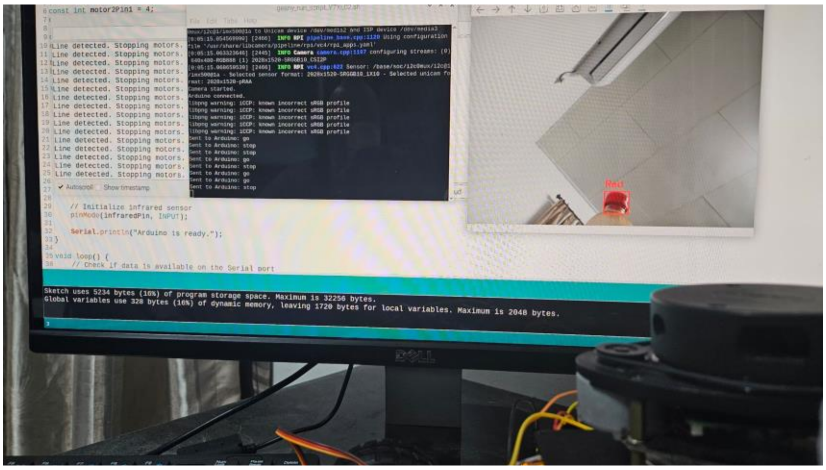


Figure 4.17 Raspberry pi and Arduino results (Walkway line detected)

4.2.8 Final assembly for track

The final configuration for assembling the track signed to be very basic but efficient enough for providing a test ground for the self-driving car. The track is made out of plywood due to its ability to withstand much wear, it provides an even surface and is easily modifiable. This material makes it possible to use the track many times without the structure deforming or changing its outward appearance.

The track design adopt simplitrack in design to accommodate instructional activities as well as for experimental purposes. We by design had incorporated only one stop sign and a mock traffic signal in the form of a green light whereby the car is trained to identify traffic lights and signs. We place it particularly at the strip to ensure that the stop sign and the traffic light are positioned rightly to ensure the genuine atmosphere of work on the strip.

The size of the track was standardized at the ratio of 1/10 of the real size of the Malaysian roads in order to realistically mimic real conditions on the track. The width and length of the track are 62mm and 1000mm respectively; the width stands for the scaled-down width of a normal two way highway in Malaysia. This commitment to scale improves the realism and pedagogical benefit of the undertaking.

This assembly process was simple and considering the need of every component, they were well placed to suit the course of the process and the attention of the subject matter. Due to the low complexity of the track design, it can be easily transported as well as installed in different educational environment. All in all, the last process of track assembly is an effective and feasible solution to the problem of practicing the autonomous car and stimulating the students' interest through the effective simulation.

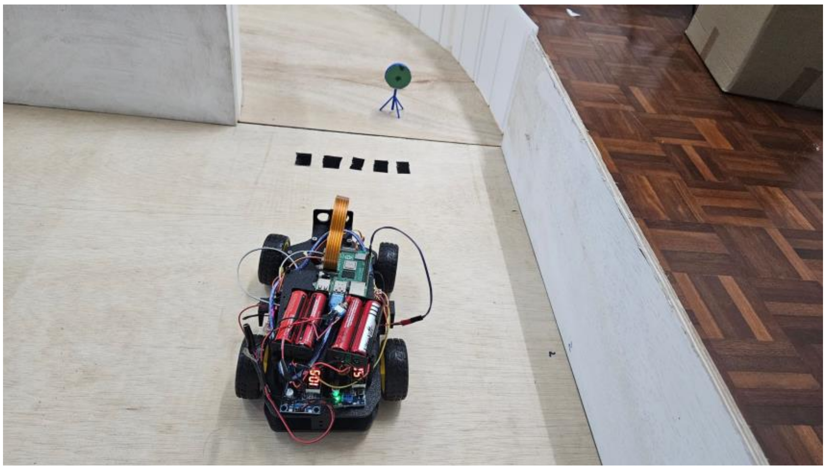


Figure 4.18 Walkway, green traffic light

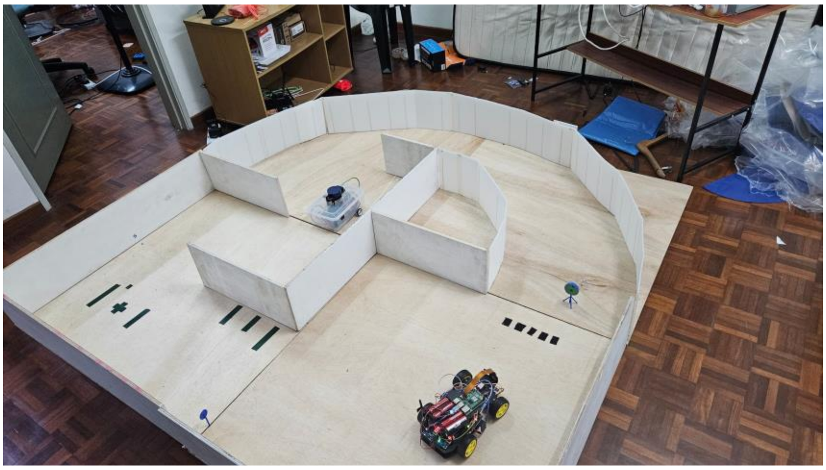


Figure 4.19 Full track assembly with the cars

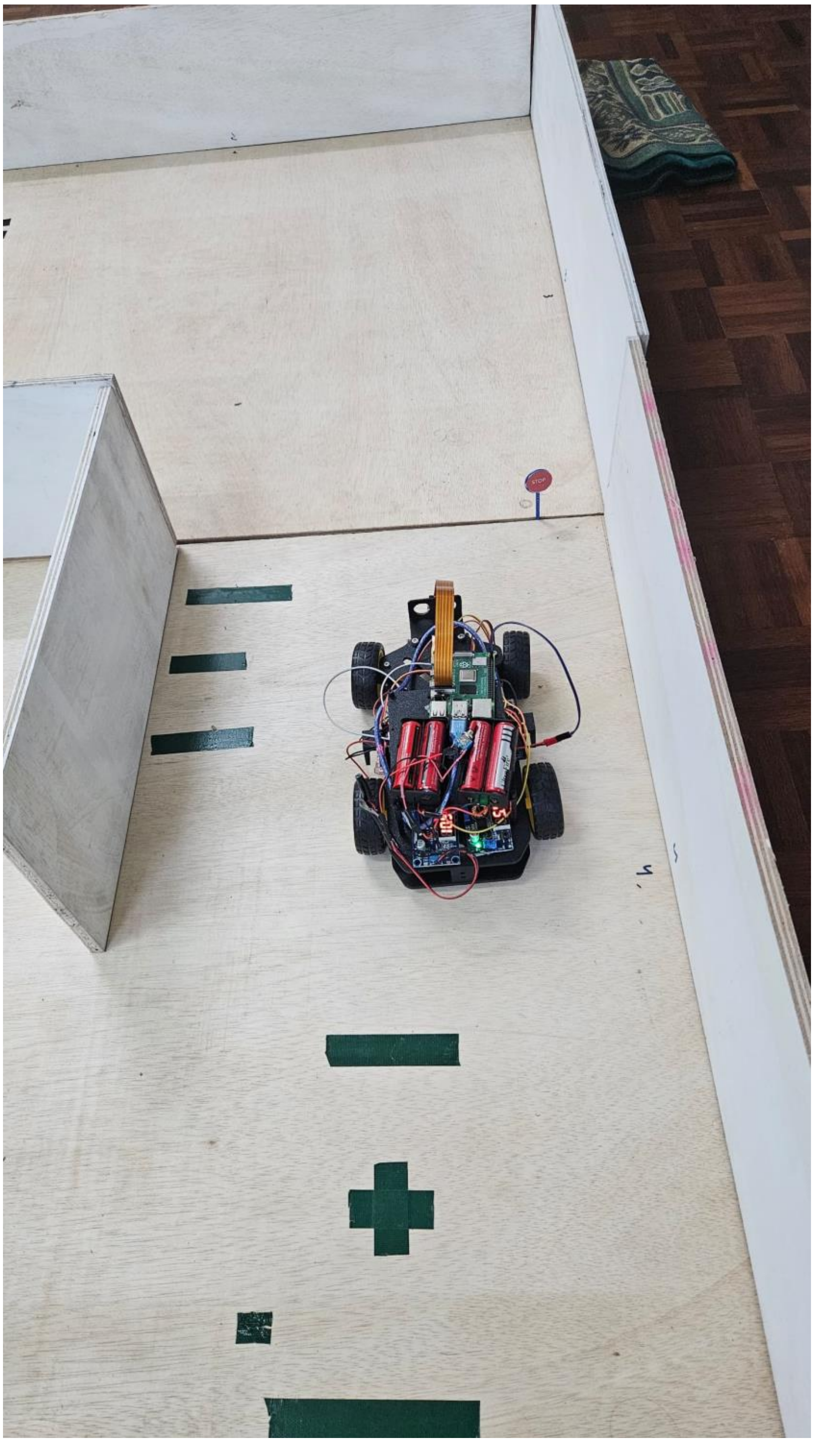


Figure 4.20 Stop sign

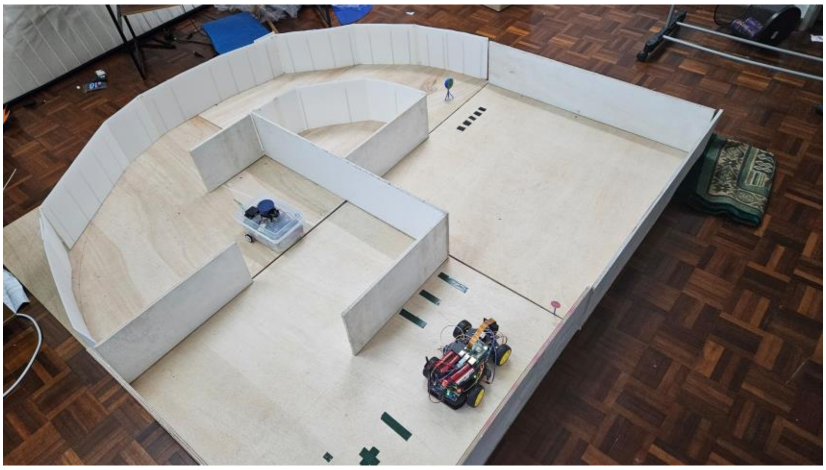


Figure 4.21 Full track assembly with the cars another angle

4.3 KTP (Knowledge Transfer Program

The concept of KTP is an attempt to undertake a practical transfer of knowledge underpinned by a belief that gaps exist between academia and practice, and it targets the young generation. In the framework of this standing, we have recently paid a school visit to raise young people's awareness of autonomous vehicles. The objective of this educational session was to create an atmosphere of invention and knowledge concerning advanced technology and present knowledge of it in an attractive manner to the kids.

In a live session today, we showcased a basic autonomous car model which uses an ultrasonic sensor for obstacle sensing. This basic model of the car was also set up with a function that enabled remote control through a Smartphone, introducing an aspect of car remote control and automation. The students were very active, graspping more knowledge on how the sensors operate when detecting obstacles and their application when controlling the movement of a vehicle.

Besides the current model, future development related to the track and the vehicle will be in store for us. In the new version additional more advanced technologies will be foreseen, like cameras or LiDAR systems. These elements will not only enhance the car's capatrack to operate independently on the roads but also will enhance the learning outcomes in the process. By incorporating computation vision in cars through cameras and depth sensing using LiDAR, the enhanced cars will be able to see their surrounding environment just as self-driven automobiles do by making use of actual sensors.

This ongoing project aims to provide a comprehensive learning platform for students, enabling them to gain hands-on experience with cutting-edge technology. By exposing young minds to these innovations, we hope to ignite their interest in the field of robotics and autonomous systems, encouraging them to pursue further studies and careers in this rapidly evolving sector. The practical experience gained from interacting with such projects could serve as a valuable foundation for their future education and professional endeavors.

Figure 4.22 KTP in school

4.3.1 Track Design

In the process of planning an entertaining and informative extravaganza for the Knowledge Transfer Program (KTP), my models include a sellable product – a race track – made in SolidWorks. The track is planned to be as modular as possible to accommodate changes in the acceptance model and adapt the learning activities. This flexibility will allow making the layout changes very faster whenever we want to demonstrative some exercises which may be the simple obstacle detection or any other thing that requires the car to navigate on the path all by its own.

In an effort to familiarly captivate the students as well as make the track as realistic as possible, I have introduced a number of components on the track. It's now possible to identify racing lines including the start and finishing lines, lanes and turns. These are not only visually appealing but are also useful for helping to steer the self-driving cars and for helping to indicate the path to follow in line-following algorithms during demos.

Also, I have placed few small buildings around the track modeled in 3D to provide a like feeling as in the real environment of an urban track. These structures include simple models such as houses; traffic lights; and signs and this make it easier to understand the lesson as the teacher demonstrates it because what he is demonstrating is actually real life real like a mini track. Such arrangement assists the students in understanding how the self-driving cars operate in real life and how they go about their proceedings regarding the environment in which they exist.

With a view of deepening the participation of the students I am planning to incorporate other phases of the track to be lit, mini berths for the audience to sit as if watching a race event. By incorporating these realistic features, the track will not only be an attractive teaching aid and imaginative play area but will also allow the children to become interested in robotics,engineering and self-driving cars. This practical way will assist in helping the teacher in making topics easy and interesting for the young students.

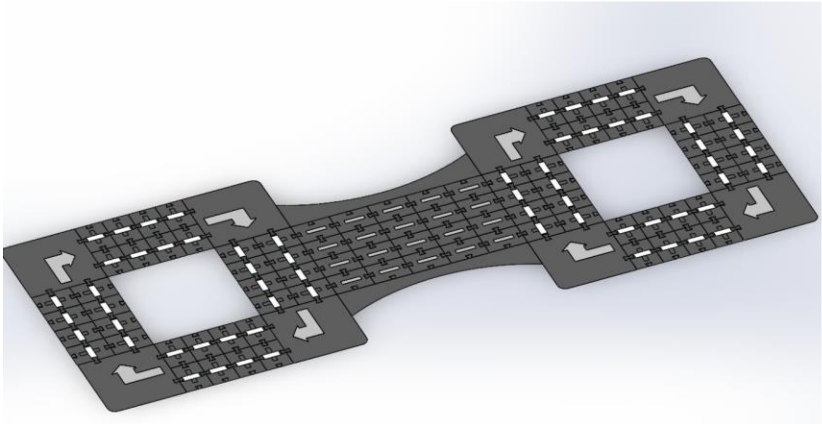


Figure 4.23 Solidworks track design

Figure 4.24 Final assembly of track

CHAPTER 5

CONCLUSION

5.1 Research Outcomes

Thus, this work is completed and produced considerable research findings and it was achieved objectives formulated at the start. The first aim of the research was to build a 1/10 car model system adapting the OTORAYA Autonomous Vehicle (OAV) to the YOLO application. This objective was achieved with meticulous planning and execution and thus created a functional and realistic track. It is of Malaysian standard, whereby the track was built from plywood and was fully constructed following correct procedure. A stop sign was placed on the wall, a green traffic sign was installed, and accurate lane lines were incorporated so that it offers a comprehensive test environment. The uncertainty is further compounded by the construction of the track – the track is made in modular format and can be easily assembled and taken apart – this makes it ideal for demonstrations and presentations. This track gives physical implementation of actual world situations which in turn will help the correct testing of AV functionality at appropriate scale.

The second goal was to create an unmanned automobile model, specifically a 1/10 scale CAMERA MODEL OTORAYA, based on the YOLO application . The aim of this goal was met through the implementation of evolution technologies including Raspberry Pi, AI camera, Arduino Uno and related components. All in all, it can be concluded that the vehicle also successfully performed the identification of its environment and its reaction to it. Consequently, the system designed for object detection will use the YOLO algorithm for traffic light, sign, and lane line detection and similar objects in real time. The Raspberry Pi analyses the visual input and issue the relevant instructions to the Arduino regarding the motors and servo system. For example, when it identifies a green traffic light, the Raspberry Pi sends a signal to the Arduino board that says, ‘Go” and the car can continue moving. Likewise, whenever a red light is identified, the system ensures that the car only stops when it identifies the location of the walkway line, thereby addressing the problem of depth estimation without an accompanying depth camera. The outcomes of the project also prove that the track and the vehicle designed is competent and accurate. This scaled model possess a dynamic behavior in recognizing stop sign, density and traffic lights and lane detection through an infrared sensor. Altogether, the track and vehicle give an integrated low-profile platform that is practical, scalable, and educative.

These successes underscore the impact of the project and broaden it out. The realistic track system along with a functional autonomous car model give a prove useful in research, training and skill enhancement in the study of autonomous vehicles. The current work lays the foundation for future extensions, including the addition of more sensors, improved algorithms for localization, and real-time processing of data to create even more applications for autonomous systems.

5.2 Future Recommendation

Today's success of the OTORAYA Autonomous Vehicle (OAV) and its related track laid down the capability that can be improved added. However, there are several areas in the system that can be enhanced to improve significantly the performance, accuracy and functionality of the system.

One would be that the device needs more powerful processor for it as the current one is not that powerful. Such a Raspberry Pi utilized in the current design is sufficient for executing simple tasks; nonetheless, using a Raspberry Pi 5 or, preferably, an NVIDIA Jetson Nano boosts the system's performance. These devices have enhanced performance capatrack in terms of capability to solve algorithms with better efficiency and better producing systems for object detection, this hence lowers the response time.

Another critical enhancement is the possibility of using depth cameras as Intel RealSense or ZED Mini. To date, the system uses a standard camera to monitor the environment, but that is a definitive limitation since it has no depth of field. An integration of a depth camera would enhance the capabilities of the vehicle to measure distance in a better way and thus navigate and avoid objects. This would also do away with having to make do with features like line detection for stopping at some predetermined points as not as accurate as in real life emulation.

Use of higher end LiDAR or ultrasonic sensors would increase the range capability of the sensor system and the range at which the vehicle sees obstacles in the environment. Although the line detection was provided by infrared sensors, LiDAR on this vehicle would provide high-resolution mapping of the environment, enhancing safety and the ability to respond in diverse conditions. All these sensors could easily be incorporated into the existing system for a much more effective autonomous system.

The track itself could also use improvements, for instance, the addition of improved objects such as intersection, pedestrian crossings or even traffic signals. These features would set the driving conditions to be more real and present a challenging experience to the car. Also there could be improvements in the realism of lighting systems, texture of the roads and physical barriers that are possible to move in order to create a better testing track.

However, increasing the communication level between the Raspberry Pi and Arduino would also be desirable. Although the current system applies one of the most primitive forms of communication, shifting toward a faster, such as the I2C or CAN bus protocol, would lead to even better response times, which are likely to be required as more sensors and components are incorporated in the system. This would make it easier and faster to operate and resolve matters arising from the operational real time applications.

In the context of machine learning, generalizing the already existing YOLO application to provide newer classification algorithms will allow the car to discern a lot of other objects including people in cross walks or complicated road sings. Such functions would enable the vehicle to enhance its learning ability so that it would function optimally in the various environments possible. This could be helpful especially in real life tests since new cases are likely to emerge at any one time.

There would also be a need to improve on power management. Now, the battery pack for the vehicle and for powering other facilities and components is incorporated; however, replacing it with a new rechargeable LiPo battery type with a greater capatrack will enhance the runtime and stability of the system. A possible method of maintaining the power supply to be as efficient as possible for performance and ‘safe’ was to introduce battery management system (BMS).

Another improvement that may be incorporated into the system is the Wireless Monitor and Control. As with Wi-Fi or Bluetooth control and monitoring would become possible and real-time, users would be able to watch camera streams, modify vehicle parameters, and analyze sensor data on the go. This would further be helpful for the user and enhance the understanding about the performance of the vehicle.

Last, a component-based approach to design the system would provide more opportunities for updates in the future. Since the track can be built out of modular sections and the individual components of the vehicle can be replaced without it becoming impossible to redesign the track to accommodate them, both the vehicle and the track could be updated to reflect improvements in technology. Moreover, it may be helpful to incorporate in the system an enhanced information section, like a application or GUI, that can provide a full background on the concept of autonomous vehicles to the users including students.

If implemented, applying these recommendations will enhance the operation and efficacy of the OTORAYA Autonomous Vehicle and track; it will also serve to present the system as a more extensive and progressive means for research, experimentation, and training in the growing specialty of autonomous vehicle science. These improvements would make it all more useful and practical to everyone involved in learning or teaching and a perfect real life tool.

REFERENCES

| Graphic illustration of Burgess' concentric zone model. Nb: Zones 4... | Download Scientific Diagram. (n.d.). Retrieved June 10, 2024, from https://www.researchgate.net/figure/Graphic-illustration-of-Burgess-concentric-zone-model-Nb-Zones-4-and-5-Rubenstein_fig2_337863057

02_Guidelines_Geometric_Standards_On_Roads_Networks_System. (n.d.).

Arkadiusz, M., & Biel, S. (n.d.). *The Investigation into the use of YOLO on overlay images for the purpose of privacy protection*.

Barcelona Smart Track: most remarkable Example of Implementation. (n.d.). Retrieved May 21, 2024, from https://www.e-zigurat.com/en/blog/smart-track-barcelona-experience/

Bartneck, C., Lütge, C., Wagner, A., & Welsh, S. (2021). Autonomous Vehicles. In *SpringerBriefs in Ethics* (pp. 83–92). Springer Nature. https://doi.org/10.1007/978-3-030-51110-4_10

Batty, M. (2022). The Linear Track: illustrating the logic of spatial equilibrium. *Computational Urban Science*, *2*(1), 1–17. https://doi.org/10.1007/S43762-022-00036-Z/FIGURES/12

Bonnefon, J. F., Shariff, A., & Rahwan, I. (2019). The trolley, the bull bar, and why engineers should care about the ethics of autonomous cars. *Proceedings of the IEEE*, *107*(3), 502–504. https://doi.org/10.1109/JPROC.2019.2897447

Buganza Co-Supervisors, T., Trabucchi, D., & Magistretti Daniel Ramos Sampaio, S. (n.d.). *Towards Disruption of the Automotive Industry: the Emergence of Different Approaches*.

Evans, K., de Moura, N., Chauvier, S., Chatila, R., & Dogan, E. (2020). Ethical Decision Making in Autonomous Vehicles: The AV Ethics Project. *Science and Engineering Ethics*, *26*(6), 3285–3312. https://doi.org/10.1007/s11948-020-00272-8

How Does a Self-Driving Car See? | NVIDIA Blog. (n.d.). Retrieved May 5, 2024, from https://blogs.nvidia.com/blog/how-does-a-self-driving-car-see/

Hub-and-spoke office model. 4 reasons why it matters - ShareSpace Blog. (n.d.). Retrieved June 10, 2024, from https://www.sharespace.work/blog/en/expert-article-en/hub-and-spoke-4-reasons-why-it-is-a-business-model-you-should-think-about/

Hübner, D., & White, L. (2018). Crash Algorithms for Autonomous Cars: How the Trolley Problem Can Move Us Beyond Harm Minimisation. *Ethical Theory and Moral Practice*, *21*(3), 685–698. https://doi.org/10.1007/s10677-018-9910-x

Hussain, R., & Zeadally, S. (2019). Autonomous Cars: Research Results, Issues, and Future Challenges. In *IEEE Communications Surveys and Tutorials* (Vol. 21, Issue 2, pp. 1275–1313). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/COMST.2018.2869360

Inside Masdar Track, the $22bn eco-project in the Arabian desert. (n.d.). Retrieved May 2, 2024, from https://www.thetimes.co.uk/article/inside-masdar-track-the-22bn-eco-project-in-the-arabian-desert-rb7vcrhlv

Kumar, A., & Mehar, V. (2023). Available at www.rjetm.in 72. In *Research Journal of Engineering Technology and Medical Sciences* (Vol. 06, Issue 01). www.rjetm.in

Levels of autonomous driving according to SAE J3016. | Download Scientific Diagram. (n.d.). Retrieved May 5, 2024, from https://www.researchgate.net/figure/Levels-of-autonomous-driving-according-to-SAE-J3016_fig1_322368082

Martin, G. (2019). Sustainability prospects for autonomous vehicles: Environmental, social, and urban. In *Sustainability Prospects for Autonomous Vehicles: Environmental, Social, and Urban*. Taylor and Francis. https://doi.org/10.4324/9781351109956

Motwani, N. P., & S, S. (2023). Human Activities Detection using DeepLearning Technique-YOLOv8. *ITM Web of Conferences*, *56*, 03003. https://doi.org/10.1051/itmconf/20235603003

Othman, K. (2021). Public acceptance and perception of autonomous vehicles: a comprehensive review. *AI and Ethics*, *1*(3), 355–387. https://doi.org/10.1007/s43681-021-00041-8

Out of Two Million People, Most Prefer That a Self-Driving Car Kill the Elderly. (n.d.). Retrieved May 5, 2024, from https://www.popularmechanics.com/technology/infrastructure/a24222017/trolley-problem-self-driving-cars/

Parekh, D., Poddar, N., Rajpurkar, A., Chahal, M., Kumar, N., Joshi, G. P., & Cho, W. (2022). A Review on Autonomous Vehicles: Progress, Methods and Challenges. *Electronics (Switzerland)*, *11*(14). https://doi.org/10.3390/electronics11142162

Singapore is a smart track: learn why - We Build Value. (n.d.). Retrieved May 21, 2024, from https://www.webuildvalue.com/en/megatrends/singapore-smart-track.html

Smart Cities World - Cultural space - Smart track futures showcased in Dubai. (n.d.). Retrieved May 21, 2024, from https://www.smartcitiesworld.net/cultural-space/smart-track-futures-showcased-in-dubai-8250

Smart Track Amsterdam - About Smart Cities®. (n.d.). Retrieved May 21, 2024, from https://www.aboutsmartcities.com/amsterdam-smart-track/

The six block clusters and the four street clusters. The different... | Download Scientific Diagram. (n.d.). Retrieved June 10, 2024, from https://www.researchgate.net/figure/The-six-block-clusters-and-the-four-street-clusters-The-different-types-belong-to-the_fig4_256895610

The Use of RADAR Technology in Autonomous Vehicles | System Analysis Blog | Cadence. (n.d.). Retrieved May 5, 2024, from https://resources.system-analysis.cadence.com/blog/msa2022-the-use-of-radar-technology-in-autonomous-vehicles

Washington, D. C. 's, & Diehl, S. B. (2018). *RETROFITTING THE HIGH DENSITY TRACK FOR AUTONOMOUS VEHICLES*.

Yolo V8: A Deep Dive Into Its Advanced Functions and New Features | by Mujtaba Raza | Medium. (n.d.). Retrieved May 2, 2024, from https://medium.com/@mujtabaraza194/yolo-v8-a-deep-dive-into-its-advanced-functions-and-new-features-f008599fe604

Zamindar, A. (n.d.). ARTIFICIAL INTELLIGENCE IN SELF-DRIVING CARS RESEARCH AND INNOVATION. *Www.Irjmets.Com @International Research Journal of Modernization in Engineering*, *889*. www.irjmets.com

+++++++++++++

APPENDIX

Appendix-A

Code

Raspberry Pi code:

```
import cv2
import numpy as np
from picamera2 import Picamera2
import serial
import YOLO from ultralatics
def detect_road_lines(frame):
    """
    Detect road lines in the frame and determine the steering direction.
    """
    # Convert to grayscale
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Apply Gaussian blur
    blurred = cv2.GaussianBlur(gray, (5, 5), 0)

    # Perform edge detection
    edges = cv2.Canny(blurred, 50, 150)

    # Region of interest mask
    height, width = edges.shape
    mask = np.zeros_like(edges)
    roi_corners = np.array([[
        (0, height),
        (width // 2 - 50, height // 2 + 50),
        (width // 2 + 50, height // 2 + 50),
        (width, height)
    ]], dtype=np.int32)
    cv2.fillPoly(mask, roi_corners, 255)
    cropped_edges = cv2.bitwise_and(edges, mask)

    # Hough Line Transform
    lines = cv2.HoughLinesP(cropped_edges, 1, np.pi / 180, 50, minLineLength=50,
maxLineGap=150)

    # Analyze line angles
    direction = "straight"
    if lines is not None:
        left_lines, right_lines = [], []
        for line in lines:
            x1, y1, x2, y2 = line[0]
            slope = (y2 - y1) / (x2 - x1) if x2 != x1 else np.inf
            if slope < 0:
                left_lines.append(line)
            else:
                right_lines.append(line)

        if len(left_lines) > len(right_lines):
            direction = "left"
```

```
        elif len(right_lines) > len(left_lines):
            direction = "right"

    return frame, direction

def detect_stop_sign(frame, stop_cascade):
    """
    Detect stop signs in the frame.
    """
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    stop_signs = stop_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5,
minSize=(30, 30))
    for (x, y, w, h) in stop_signs:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255), 2)
        cv2.putText(frame, "Stop", (x, y-10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 0, 255),
2)
    return frame, len(stop_signs) > 0

def display_camera_with_line_and_sign_detection(picam2):
    """
    Display live camera feed with road line detection and stop sign detection.
    """
    arduino = serial.Serial('/dev/ttyACM0', 9600)
    last_command = None

    stop_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'stop_sign.xml')

    while True:
        frame = picam2.capture_array()

        # Detect road lines
        frame, direction = detect_road_lines(frame)

        # Detect stop signs
        frame, stop_detected = detect_stop_sign(frame, stop_cascade)

        # Display the frame
        cv2.imshow("Road Line and Stop Sign Detection", frame)

        # Send commands to Arduino
        if stop_detected:
            if last_command != "stop":
                arduino.write(b'stop\n')
                print("Sent to Arduino: stop")
                last_command = "stop"
        elif direction != last_command:
            if direction == "left":
                arduino.write(b'left\n')
                print("Sent to Arduino: left")
            elif direction == "right":
                arduino.write(b'right\n')
                print("Sent to Arduino: right")
            elif direction == "straight":
                arduino.write(b'straight\n')
                print("Sent to Arduino: straight")
            last_command = direction
```

```
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    arduino.close()
    cv2.destroyAllWindows()

if __name__ == "__main__":
    picam2, config = initialize_camera()
    try:
        start_camera(picam2, config)
        display_camera_with_line_and_sign_detection(picam2)
    finally:
        stop_camera(picam2)
```

Arduino code:

```
#include <Servo.h>

// Pin definitions
const int motor1Pin1 = 5;
const int motor1Pin2 = 6;
const int motor2Pin1 = 4;
const int motor2Pin2 = 7;

const int servoPin = 10;
const int infraredPin = A0;

Servo servoMotor;
bool isStopped = false;

void setup() {
    // Initialize motor pins
    pinMode(motor1Pin1, OUTPUT);
    pinMode(motor1Pin2, OUTPUT);
    pinMode(motor2Pin1, OUTPUT);
    pinMode(motor2Pin2, OUTPUT);

    // Initialize servo motor
    servoMotor.attach(servoPin);
    servoMotor.write(90); // Neutral position (facing forward)

    // Initialize serial communication
    Serial.begin(9600);

    // Initialize infrared sensor
    pinMode(infraredPin, INPUT);

    Serial.println("Arduino is ready.");
}

void loop() {
    // Check if data is available on the Serial port
    if (Serial.available() > 0) {
        // Read the incoming command
        String command = Serial.readStringUntil('\n');
        command.trim(); // Remove any whitespace
```

```
    if (command == "go") {
      isStopped = false;
      moveMotors(true);
    } else if (command == "stop") {
      isStopped = true;
      moveMotors(false);
    } else if (command == "left") {
      servoMotor.write(45);
    } else if (command == "right") {
      servoMotor.write(135);
    } else if (command == "straight") {
      servoMotor.write(90);
    }

  }

  // If stopped, wait for the infrared sensor to detect the line
  if (isStopped) {
    int lineDetected = digitalRead(infraredPin);
    if (lineDetected == HIGH) { // Assuming HIGH means line detected
      Serial.println("Line detected. Stopping motors.");
      moveMotors(false); // Stop the motors
    }
  }


}

void moveMotors(bool forward) {
  if (forward) {
    // Move both motors forward
    digitalWrite(motor1Pin1, HIGH);
    digitalWrite(motor1Pin2, LOW);
    digitalWrite(motor2Pin1, HIGH);
    digitalWrite(motor2Pin2, LOW);
  } else {
    // Stop both motors
    digitalWrite(motor1Pin1, LOW);
    digitalWrite(motor1Pin2, LOW);
    digitalWrite(motor2Pin1, LOW);
    digitalWrite(motor2Pin2, LOW);
  }
}
```

ABSTRACT

The development uses artificial intelligence perception to construct an autonomous vehicle on a 1/10th scale that obtains Level 5 autonomy through system integration of navigation and perception capabilities. Image perception operations benefit from the accurate detection system created through the combination of OpenCV and YOLOv8. The management of motor operations functions through two systems. The Raspberry Pi controls low-level procedures but the Arduino system handles vision-based operations. The camera system records high-definition environmental information that enables automatic detection of track boundaries along with features as well as obstacle recognition with a response time under 5 seconds on average. The servo motor uses the DC motor to drive the vehicle up to 10 km/h speed with precise steering capabilities. Real-time mapping capabilities of the vehicle system when paired with ultrasonic sensors enable the system to perform proximity detection functions for secure operation. The performance testing system needed proper development of a track to simulate demanding operational scenarios including curved paths and safety obstacles in limited operational boundaries. The detection system failed to achieve accurate results when operating at low-light or during unstable routes because of integration challenges and environmental factors which limited its precision. The reliability standards of sensor fusion approaches require adaptive learning improvement to meet performance standards along with faster image processing rates that uphold those standards. The autonomous navigation project functions as a robotic control technology development while delivering practical benefits toward education for students in robotics and artificial intelligence throughout smart transportation systems. Modern advanced technologies enabled engineers to develop an autonomous system which supports the creation of self-governed urban transportation networks.