Istanbul Medipol University

School of Engineering and Natural Sciences

Graduation Project

2022-2023

PROJECT TITLE
AUTONOMOUS AI VEHICLE PROJECT BY USING 1 CHANNEL LIDAR SENSOR AND RASPBERRY PI CAMERA
PROJECT ADVISOR
Prof. Dr. Bahadır Kürşat GÜNTÜRK
TEAM MEMBERS
Enes UĞUZ

Project Code

Istanbul Medipol University

School of Engineering and Natural Sciences

Graduation Project

Project Title:

AUTONOMOUS AI VEHICLE PROJECT BY USING 1 CHANNEL LIDAR SENSOR AND RASPBERRY PI CAMERA

Project Advisor:

Prof. Dr. Bahadır Kürşat GÜNTÜRK

Project Team Members:

Enes UĞUZ

Sponsor Company (if any):

BUDGET (TL)	PROPOSED	CONSENTED
IMU FUNDING	10170	11220
SPONSOR COMPANY FUNDING		
TOTAL	10170	11220

PROJECT PLAN	PROPOSED	CONSENTED
PROJECT PLAN Duration in Weeks	28 Weeks	28 Weeks
STARTING DATE		

Project Code

Istanbul Medipol University
School of Engineering and Natural Sciences
Graduation Project

PROJECT ADVISOR	DEPARTMENT CHAIR
Name: Prof. Dr. Bahadır Kürşat GÜNTÜRK	**Name:** Doç. Dr. Mehmet Kemal Özdemir (EE) **Name:** Prof. Dr. Hakan Tozan (IE)
Contact Information: Tel : +90 (216) 681 5141 E-mail : bkgunturk@medipol.edu.tr	**Contact Information:** Mehmet Kemal Özdemir Tel : +90 (216) 681 5626 E-mail : mkozdemir@medipol.edu.tr
Signature:	**Signature:**

TEAM MEMBER	TEAM MEMBER
Name: Enes UĞUZ	**Name:**
Contact Information: Tel : +90 (553) 046 25 33 E-mail : enes.uguz@std.medipol.edu.tr	**Contact Information:**
Signature:	**Signature:**

Project Title: AUTONOMOUS AI VEHICLE PROJECT BY USING 1 CHANNEL LIDAR SENSOR AND RASPBERRY PI CAMERA
Project Advisor: Prof. Dr. Bahadır Kürşat GÜNTÜRK
Team Members: Enes UĞUZ
Project Group Title:

Istanbul Medipol University
School of Engineering and Natural Sciences
Graduation Project

PROJECT OVERVIEW/SUMMARY/ABSTRACT

The aim of this project is to create an autonomous vehicle by combining the latest technologies. In this autonomous car, the car creates a decision mechanism according to the images it receives. This decision mechanism has different variables, for example, lane detection is made with Hough transform to ensure the movement and turns of the vehicle, and the vehicle decides on the direction according to the lanes. With Object detection and YD Lidar X4 sensor, it is ensured that the vehicle stops when it encounters any obstacle or object on the road. After scanning the environment with Lidar, if any object is detected on the road, the vehicle is stopped. It is aimed to prepare an artificial intelligence model according to the data collected from the road with the joystick controller and assist the lane detection part in the direction decision. The combination of all these features makes this project very different from other autonomous vehicle projects.

There are 6 different work packs in this project. When these work packs come together, they form the whole project. The first and most important part of the project is the assembly of all the car parts. This item covers the entire hardware part of the car. means that when this work package is completed, the car will be completed as an image. The other three work packages are related to lane detection, Lidar Sensor, and object detection. These work packages are including the car's motion algorithms. The 5th work package covers the data collection required for the ai model. To make this study package, firstly, the photos will be recorded and then trained after manually driving the vehicle on the road with the joystick controller. This model will help lane tracking and make the vehicle safer. will make it. the last work package is about testing the whole system working properly.

The first 4 work packs were completed successfully, but many set back occurred in the first work package and many parts had to be changed. 5th work package could not be completed successfully because too much time was spent for the work package 1.
However, the purpose of this work package was to help the system, failure to complete this work package does not prevent the project from being completed. Also, looking at the other results, 90% of the lane detection, 90% of the final tests, and the others were completed with 100% success. Therefore, it can be said that the project is completed successfully, even if there are some errors and delays.

- **Keywords**: Autonomous systems, Artificial intelligence, Machine learning, Robotics, Navigation, Lidar sensor, Raspberry Pi camera, Jetson Nano, Vehicle control, Decision- making,

1. OBJECTIVE OF THE PROJECT:

The objective of this project is to construct and develop an autonomous vehicle that showcases the potential of autonomous vehicles in a fundamental yet promising manner.

For this project, Raspberry Pi camera module, a 1 Channel LIDAR sensor for distance measurement and environmental mapping, as well as a Jetson Nano computer to execute our model, will be utilized as the key equipment and materials. Additionally, Arduino UNO boards will play a significant role in the development process. Autonomous vehicles can be used in various places and for a wide range of use cases. Real-world use cases of such devices have been mentioned in later pages of this report.

The main objective is for an autonomous vehicle to be built, which can act according to its environment without causing any issues from point A to B. Furthermore, it is aimed for this vehicle/car to have a compact design and form factor. If this car can be built with the expected functionalities in a tight package, the size can be changed according to the purpose of the vehicle. For instance, a smaller car might be desired for research and rescue, whereas a larger vehicle might be needed for deliveries.

Upon completion of the project, a functional autonomous vehicle will be capable of making maneuvers and acting in response to its environment. This will be achieved through a combination of hardware and software, enabling intelligent decisions to be made and efficient navigation to be carried out.
The list of the objectives of this project are:

- Following the path by image processing on the path I have drawn.
- 360mapping with lidar and avoiding obstacles.
- Detection of pedestrians and stop sign with Object detection.
- so that the vehicle can react quickly according to the image, the raspberry pi camera should run at least 15fps.

2. LITERATURE REVIEW:

2.1 Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

This seminal paper presents Faster R-CNN, a groundbreaking object detection method that significantly improves the speed and accuracy of detection. The authors introduce Region Proposal Networks (RPN) to efficiently generate region proposals and perform object detection in a single, unified framework. By sharing convolutional features between the RPN and the detection network, Faster R-CNN achieves impressive real-time performance and state-of-the-art accuracy on various object detection benchmarks. This work has had a profound impact on the development of object detection algorithms and serves as a foundation for subsequent research in the field.[1]

2.2 YOLO: You Only Look Once - Unified, Real-Time Object Detection

The YOLO paper introduces an innovative real-time object detection approach that revolutionizes the way we detect objects in images. Instead of using region proposals and performing multiple passes over the image, YOLO treats object detection as a single regression problem, directly predicting bounding boxes and class probabilities from a single neural network. This design choice enables YOLO to achieve remarkable speed while maintaining competitive accuracy. By eliminating the need for complex post-processing and feature extraction steps, YOLO simplifies the object detection pipeline, making it suitable for applications requiring rapid and efficient object detection in real-time scenarios. [2]

2.3 SSD: Single Shot MultiBox Detector

SSD is a pioneering work in the field of real-time object detection. It introduces a single-shot detection paradigm that simultaneously predicts multiple bounding boxes and class scores at different scales in a single pass. By leveraging multiple convolutional feature maps with different resolutions, SSD achieves impressive accuracy across a wide range of object sizes. This unified approach not only simplifies the object detection process but also significantly reduces computational overhead, allowing it to achieve real-time performance on various hardware platforms. The paper's results have established SSD as a popular choice for real-time object detection applications in both academia and industry. [3]

2.4 Robust Lane Detection and Tracking in Challenging Scenarios

Lane detection is a critical component of autonomous driving systems, as accurate and robust lane detection is essential for safe navigation. This paper addresses the challenges of lane detection in challenging scenarios, such as poor lighting conditions, occlusions, and complex road geometries. The authors propose a robust lane detection and tracking method that utilizes advanced computer vision techniques and deep learning models to accurately identify and follow lane markings in real-world driving conditions. The approach has been extensively evaluated on various datasets, demonstrating its effectiveness in enhancing the reliability and safety of autonomous driving systems. [4]

2.5 Deep Learning-Based Lane and Road Marking Detection for Autonomous Driving

In this study, the authors explore the application of deep learning techniques for lane and road marking detection in autonomous driving scenarios. They propose a novel convolutional neural network architecture capable of automatically detecting and delineating lane markings with high accuracy. The model's robustness is demonstrated under different environmental conditions, such as varying lighting, weather, and road surface conditions. By leveraging the power of deep learning, this approach showcases significant progress in addressing the challenges of lane detection and paves the way for further advancements in autonomous driving technology. [5]

2.6 A Survey of LIDAR Technology and Its Use in Autonomous Vehicles

Light Detection and Ranging (LIDAR) technology plays a crucial role in enabling autonomous vehicles to perceive their surroundings accurately and make informed decisions. This comprehensive survey reviews the principles of LIDAR operation, different LIDAR types, and their respective advantages and limitations. Additionally, it explores the applications of LIDAR sensors in autonomous vehicles, such as 3D mapping, object detection, and localization. The paper also discusses ongoing research and future trends in LIDAR technology, shedding light on the potential for further advancements in autonomous vehicle perception systems. [6]

2.7 Lidar for Autonomous Vehicles: Recent Developments, Applications, and Outlook

LIDAR sensors are a key component of modern autonomous vehicle perception systems, providing rich 3D point cloud data for environment mapping and object detection. This paper presents recent developments in LIDAR technology, including advancements in sensor hardware, data processing algorithms, and sensor fusion techniques. It delves into the various applications of LIDAR in autonomous vehicles, such as obstacle detection, motion planning, and environmental mapping. Moreover, the paper discusses the challenges and prospects of LIDAR technology, highlighting its potential for driving significant advancements in the field of autonomous driving. [7]

2.8 End-to-End Deep Learning for Self-Driving Cars

Autonomous driving systems often require complex pipelines involving multiple modules for perception, planning, and control. In contrast, the end-to-end deep learning approach presented in this paper seeks to directly map raw sensor data to driving actions, effectively bypassing the need for handcrafted feature engineering. The authors use convolutional neural networks (CNNs) to learn a direct mapping from camera images to steering commands, enabling the car to drive autonomously in a simulated environment. Although the end-to-end approach simplifies the system, it also raises challenges related to interpretability and robustness in real-world driving conditions. [8]

2.9 DeepDriving: Learning Affordance for Direct Perception inAutonomous Driving

This research explores the concept of "affordance" for direct perception in autonomous driving. Affordance refers to the actionable information that a scene provides to an agent, enabling it to understand how to interact with the environment. The authors propose a deep learning-based approach to learn affordances directly from visual input. By incorporating affordance cues, such as road boundaries and obstacles, into the perception pipeline, the model gains a more comprehensive understanding of the environment, enhancing the decision-making process for autonomous vehicles. [9]

2.10 Learning to Drive from Simulation without Real World Labels

Training autonomous vehicles solely on real-world data can be costly and time-consuming due to the need for extensive manual annotations. To address this challenge, this paper introduces a transfer learning approach that leverages data from simulation environments to improve real-world performance. The authors propose training a deep neural network in a simulated environment where annotations are readily available, and then fine-tuning the model with limited real-world labeled data. The results demonstrate the potential of transfer learning in reducing data collection efforts while maintaining high driving performance in real-world conditions. [10]

2.11 Semantic Segmentation for Autonomous Vehicles: A Review

Image segmentation is a critical task in autonomous vehicles, as it involves classifying each pixel in an image to create a meaningful semantic map of the environment. This review article provides a comprehensive survey of semantic segmentation methods tailored specifically for autonomous driving applications. It discusses various approaches, including fully convolutional networks (FCNs), dilated convolutions, and encoder-decoder architectures, which have shown promising results in capturing fine-grained details in the scene. The paper also highlights the importance of accurate segmentation in enhancing the perception capabilities of autonomous vehicles. [11]

2.12 Failure Modes in Machine Learning for Autonomous Vehicles

Deploying machine learning models in safety-critical systems like autonomous vehicles requires a thorough understanding of potential failure modes and their implications. This paper conducts an in-depth analysis of failure modes in machine learning models for autonomous vehicles, focusing on scenarios where the system fails to make appropriate decisions or misinterprets critical environmental cues. The authors present strategies to mitigate these failure modes through rigorous testing, validation, and designing fallback mechanisms, ensuring the safety and reliability of autonomous driving systems. [12]

2.13 Legal and Ethical Aspects of Autonomous Driving

As autonomous vehicles become more prevalent, legal and ethical considerations play a vital role in shaping the adoption and regulation of this transformative technology. This paper delves into the legal and ethical aspects of autonomous driving, covering topics such as liability, responsibility allocation, privacy concerns, and societal impact. It examines the challenges faced by policymakers and researchers in creating a regulatory framework that ensures public safety while promoting innovation. The study provides valuable insights into the broader implications of autonomous vehicles on society, law, and ethical norms. [13]

2.14 Impact of Autonomous Vehicles on Traffic Flow

The integration of autonomous vehicles into existing traffic systems can have profound implications on traffic flow and overall transportation efficiency. This paper investigates the impact of autonomous vehicles on traffic flow through extensive traffic simulation models and analyses. It explores scenarios where autonomous vehicles interact with human-driven vehicles and examines how they influence congestion, travel times, and fuel consumption. The findings shed light on the potential benefits and challenges of introducing autonomous vehicles into real-world traffic environments, providing valuable insights for urban planners and policymakers. [14]

2.15 Urban Planning for Autonomous Vehicles: Challenges and Opportunities

As autonomous vehicles are poised to transform the future of transportation, this paper focuses on the implications for urban planning and city infrastructure. The authors highlight the challenges and opportunities presented by the integration of autonomous vehicles into urban environments. They explore potential changes in transportation patterns, parking needs, public transportation, and land use as a result of widespread autonomous vehicle adoption. The study emphasizes the importance of proactive planning to create cities that can harness the benefits of autonomous vehicles while addressing potential issues. It provides valuable insights for urban planners, policymakers, and city officials aiming to adapt their infrastructure to accommodate autonomous vehicles effectively. [15]

3. ORIGINALITY OF THE PROJECT

The integration of various advanced technologies the originality of this project. The combination of lane detection, object detection, lidar, and the trained model forms a comprehensive system. This integration emphasizes the project's originality and presents a holistic approach to autonomous vehicle navigation.

The project involves the use of a specifically trained model for this purpose. Training the model to interpret visual data captured by the raspberry camera module enables it to make informed decisions. This aspect of the project highlights its originality by focusing on the utilization of a customized model that caters to specific requirements. The ability of Arduino cars to autonomously navigate on a self-drawn road distinguishes the project from others. This feature requires the vehicle to adapt to a non-standard and unpredictable terrain.

Also, there is no other project where object detection and lidar are used together. There are those who use object detection and lane detection together in other projects, or there are those who move the car only with lidar, but in total such a system is not available in other projects.

In addition, another originality is that road changes can be made. In other projects, the vehicle can be tested with only one road, and this allows the vehicle to be tested only with that road. However, in this project, various road maps can be created on the road created with the puzzle system.

4. SCOPE OF THE PROJECT AND EXPERIMENTS/METHODS

4.1 JOYSTICK CONTROLLER:

At the very beginning of the project, it was desired to control the car with the joystick because it could be used to train the car later. For this, first, 2 nRF24L01were used, one of which is the receiver and the other is the transmitter. I’ve used one on the joystick shield and attached the other to the Arduino used for the vehicle.

The nRF24L01’s couldn’t connect to each other for a long time. At first, I thought there was a problem with the codes, but after a few tries, I found out that the problem was caused by nRF24L01. As a result of my research later, I discovered that it has an adapter and protects the nRF24L01from burning. After ordering the adapter, I did not have any burning problems and could use it easily.

After the burning problem was solved, the car could be moved. While the x and y values of the joystick were 0-680 on one joystick, they were 0-1020 on another joystick. Tried different codes to make the car move smoothly. For example, I tried to accelerate the car according to the angle on the joystick and the magnitude of the vector. but since the difference in speed values in the Arduino engine, does not have much effect on the vehicle, I gave up on this decision. Since the angle between the turns of the road is always constant, giving certain speeds to the engines during the turn provides an easier turn.

4.2 RASPBERRY CAMERA MODULE:

Since jetson nano and raspberry camera module are compatible with each other, I can easily use the camera module when I plug in the jetson nano raspberry camera. Once it did not work due to the cable, but when I renew the cable, I can use the camera.

I get images from the camera with GStreamer, and we have the chance to manually change the camera's properties.

capture width:1280, capture height: 720, display width:640, displayheight:360 because this way I can capture the maximum fps rate

I can get 60fps when only lane detection code works, 25fps only when object detection code works. However, when I combine both lane detection and object detection codes, the fps drops to 15 because I use a single camera.

4.3 LANE DETECTION :

Hough transform has been used for lane detection. Hough transform is a technique used in image processing and computer vision. It is used to detect and extract geometric shapes, particularly lines, from an image. The transform works by converting the Cartesian coordinate representation of a line into a parameter space, known as the Hough space. This allows lines to be represented as points in the Hough space, making it easier to detect them using voting-based algorithms. The Hough transform is widely used in applications such as line detection, lane detection, and shape recognition.

A few changes were made in lane detection. Since one of them is a circular rotation, it was a little difficult to detect this rotation with Hough transform. Because with Hough transform, I can usually detect lines. Then I tried using circle Hough transform for turning, but since the road was not that big, it was a little difficult to make the vehicle turn according to the circles I perceived, but then the old method was returned.

Another change is that I calculated the midpoint of the lanes first and then drew a line from the midpoint of the camera to the midpoint of the lanes and tried to move the vehicle according to the slope of that line. When I sent the slope to the Arduino, the speed changes in the vehicle were not so much because the speed differences in the Arduino were not that sensitive.

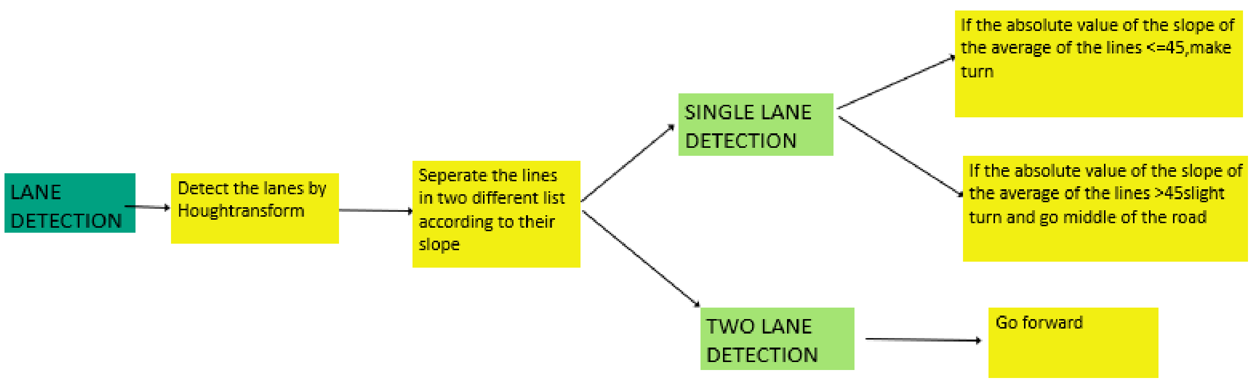

Figure-1

Figure-1 shows the model of the lane detection. According to this model, we first detect the lanes using the Hough transform. We add the detected lanes to 2 different lists according to their slope. If their slope is positive, it belongs to the right lane. If its slope is negative, it belongs to the left lane.

The reason we collect it in the list is to ensure that it does not give commands for each line and to make the decision more reliable. Direction decision is provided according to the average of the list every 0.2 seconds.

For example, if the left lane list is empty, it means that the vehicle is close to the right lane or above the right lane. and if the average of the right lane list is greater than 45, it means that the vehicle is on a straight road, and we need to center the road by making a slight left turn. However, if the average slope of the right lane list is less than 45, it means
that the vehicle must make a left turn.

Figure -2: Canny edge of the turning left

Figure -3: Houghline of the turning left

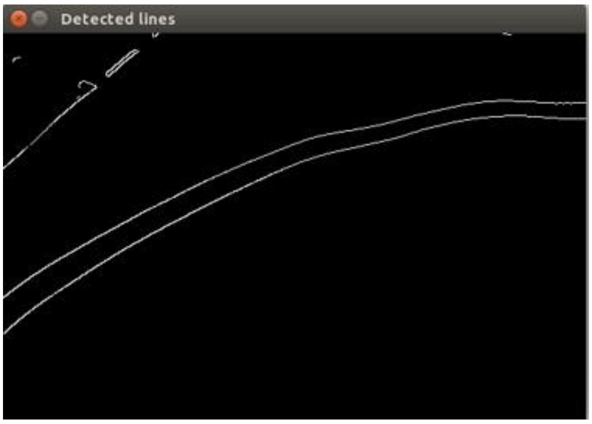

Figure -4: Canny edge of the turning right

Figure -5: Houghline of the turning right

Figure -6: Canny edge of the slight right

Figure -7: Houghline of the slight right

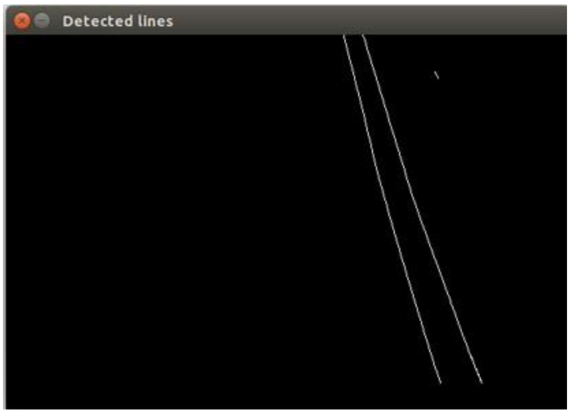

Figure -8: Canny edge of the slight left

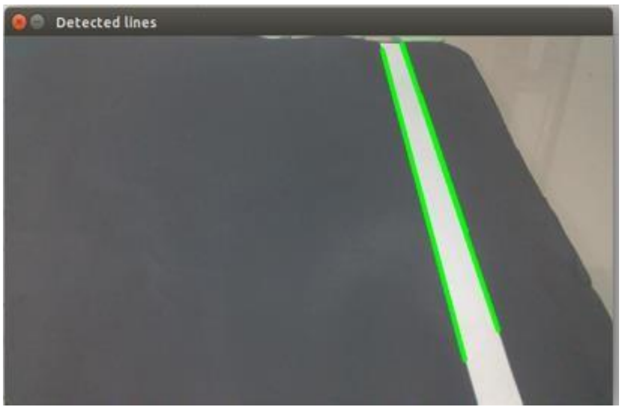

Figure -9: Houghline of the slight left

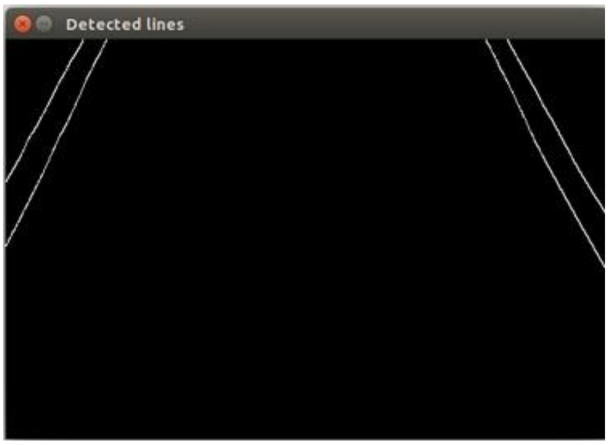

Figure -10: Canny edge of the going forward.

Figure -11: Houghline of the going forward.

4.3 CHOOSING THE COMPUTER:

Jetson nano was given in the first days of the project, and we started using jetson nano. However, jetson nano had a lot of lags and I thought it was a problem and when I talked to the assistant, I've changed the SD card because the SD card was out of memory, and it started working properly.

After completing the joystick controller, I've tried to power the jetson nano with the battery, but I could not run it with the battery even though we gave the desired volt and amperage. I started using jetson Xavier because of the battery problem, there was no power problem in the jetson Xavier because while Xavier could work between 9v-19v, but jetson nano works with 5v.

A 22.4volt battery was given by the lab for jetson Xavier. Since the battery is so heavy, a few parts were prepared with 3d modeling so that the car could lift the battery and balance the center of gravity. A long time was spent drawing these parts. Afterwards, we decided to use another battery because the battery damaged the gears of the motors, and it was very difficult in the turns due to the weight.

The vehicle was powered by batteries and lane detection was possible.
However, when I tried to run the object detection models yolov3 and yolov3-tiny, the fps was between 0-5 and this meant that the system would not work properly. Xavier could not work at full performance because OpenCV could not be installed compatible with CUDA. So, we decided to switch to jetson nano again. Because I could run object detection and lane detection codes together in jetson nano

4.4 OBJECT DETECTION

It was decided to use yolov3 for object detection at first, but then when I tried yolov3 with jetson Xavier, the fps was very bad. Then I've tried yolov3-tiny, another extension of yolov3, but it didn't give the desired performance. So, I had to choose an uncomplicated model. Since I don't need many labels in the project, I decided that this model is correct one. Due to the battery problem in the Jetson, I can run the system, but due to undervoltage, the system turns itself off after the object detection code runs. Therefore, the car will be moved using the power supply during the demo.

SSDMobileNetV2 is a deep learning model used for tasks such as object detection and image classification. It is an extension of the original SSDMobileNet model, incorporating improvements from MobileNetV2, a lightweight neural network architecture.

MobileNetV2 is designed to provide efficient and accurate feature extraction while minimizing computational resources and model size. It uses depth wise separable convolutions, which split the standard convolutional operation into separate depth wise and pointwise convolutions, reducing the number of parameters and computation.By combining the efficient feature extraction capabilities of MobileNetV2 with the object detection framework of SSD, SSDMobileNetV2 achieves a good balance between accuracy and efficiency.

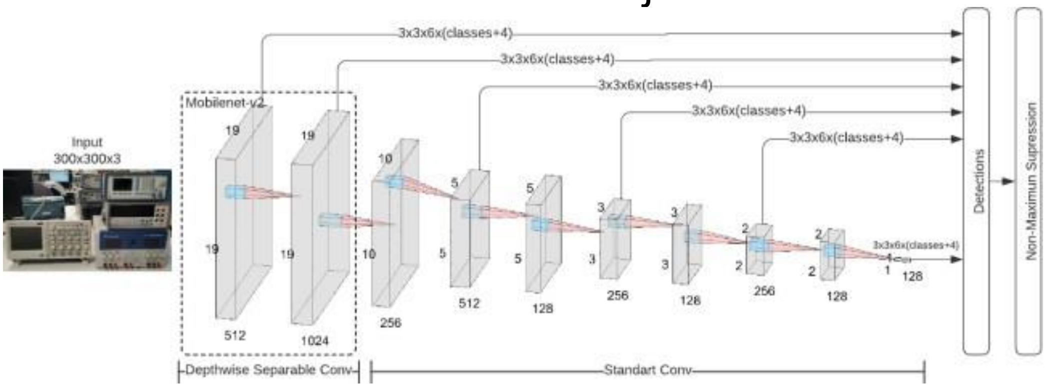

Figure-12: Architecture of the SSD Mobilenet-v2

Model Name	Speed (ms)	COCO (mAP)	Output
ssd_mobilenet_v1_coco	30	21	Boxes
ssd_mobilenet_v2_coco	31	22	Boxes
ssd_inception_v2_coco	42	24	Boxes
faster_rcnn_resnet101_coco	106	32	Boxes

Figure-13: Comparison of SSD Mobilenet-v2 with other models

4.5 YDLIDAR X4:

YDLIDAR X4 is a popular LiDAR (Light Detection and Ranging) sensor commonly used in robotics and automation applications. It provides accurate distance measurements by emitting laser beams and measuring the time it takes for the beams to return after hitting objects in the surrounding environment.

The YDLIDAR X4 offers a 360-degree scanning range, allowing it to capture a comprehensive view of its surroundings. With a detection range of up to 10 meters and a scanning frequency of 6,000 times per second, it can quickly and reliably generate detailed point cloud data.

In this project, we aim to detect the obstacles on the road by scanning the environment with lidar and not to hit these obstacles. Connection between the jetson nano and Lidar was made with USB. Then we can access the lidar with the PyLidar3 library in python. However, since we put the lidar in front of the vehicle and we only need to control the obstacles in the road, we do not need to use 360 degrees. We only need to control the angles between 120-240. In lidar, only the distance and since we perceive an angle, we cannot directly obtain an object and its width.

I wrote a code to detect the width of the object. In this code, we create a group starting from the starting angle, and then if the distance of the current angle and the adjacent angle is less than 10mm, we include it in the same group and we do this for all angles and check if there is an object. The first element of the group list we created is the starting angle of that object and the last element is the ending angle of the object. The values in between represent the distances. By using the cosine theorem, the width of the object is obtained by using the starting and ending angles and the distance of the object.

The width of the object is used to stop when it sees the obstacle. However, the width of the object is not enough to go around the object. Because the position of the object and the position of the vehicle are also important. We can learn the position of the vehicle from lane detection. and we can learn the position of the object from the starting point of the object.

For example, if the car is on the right lane and the starting angle of the object is between 170-180, the object is on the right. However, if the starting angle of the object is between 130-140 when the car is on the right lane, the object is on the left side of the road. According to the positions of the object and the vehicle, it is decided in which direction the vehicle should go around the object or stop.

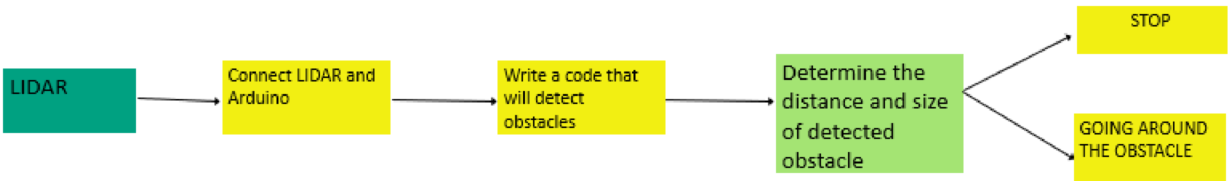

Figure-14 model of the LIDAR

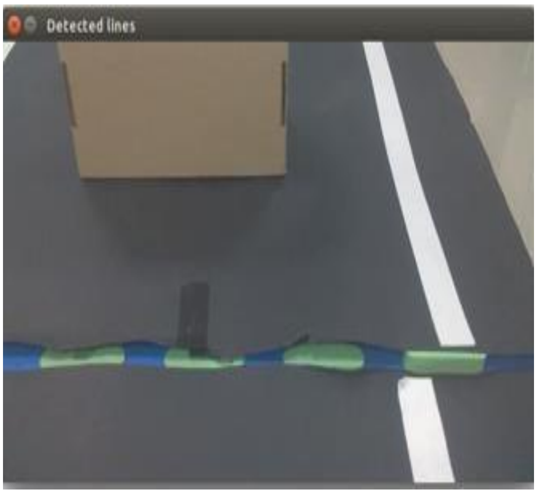

Figure-15: Obstacles image on raspberry camera module

Figure-16: plotting the obstacle in figure-10

5 PROJECT TARGETS & SUCCESS CRITERIA:

Work package 1: Successfully building the car with all the modules included.

All equipment should be placed in the car, setup should be done, circuits should be prepared, and the car should be running properly. All equipment is now installed, and the car runs well.

Work package 2: Detect the Lanes by the lane detection.

It is necessary to detect lanes with Hough transform and decide whether the vehicle should go straight or make turns according to the detected lanes.

In addition, since real-time detection is performed, most raspberry camera module should work with at least 15-20fps. Car should successfully complete 6 out of 10 laps.

The car detected the lanes and was able to turn according to the lanes. it successfully completed 7 out of 10 laps, and when just lane detection code worked it was able to get 60fps but when combined with object detection it got 20fps.

Work package 3: Avoiding obstacles by LIDAR.

The lidar should be scanned and the lidar should make the car stop when an obstacle is placed on the road. The lidar must stop at at least 8 of the 10 obstacles.

Scanning with lidar was successful and stopped at 9 out of 10 when it saw the obstacle. Also, an algorithm was written for the lidar not only to stop but also to go around the object and it was successful 4 out of 10. It is more difficult to do this because the position of the object and the position of the car are also important and speed settings are considered to be extra important, this rate is also very successful.

Work package 4: Detection of pedestrian and stop sign with object detection.

By detecting an object with Yolov3, the vehicle should be stopped when the stop sign and pedestrian are seen. FPS should be at least 15-20.

While using Xavier, object detection was done with yolov3, but the fps rate was about 0-5, and this rate was only when the object detection code was running. We must combine the lane detection and object detection codes and run it like that because only 1 camera is used. In jetson nano, ssdmobilenet-v2, which is both faster and has higher fps rate, was used. It is more suitable because it is compatible with the jetson-inference library and is a pretrained model. Also, although it works with lane detection, we get 15-20fps. detected and the vehicle was successfully stopped.

Work package 5: Gathering footage.

For the ai part of the vehicle, photos must be taken with the joystick controls and then the training of the vehicle must be completed with a model.

Work package 6 :Test the algorithms and make adjustments

After all parts are completed separately, the codes must be combined and tested as a whole, and final checks must be made. The vehicle must complete at least 6 of the 10 laps.

The car works fine when we combine all code and final adjustments made like the setting motor speeds, choosing the best framerate.

6 RISKS AND B PLANS:

Table 1: Risks and Risk Management (B-Plans)

Work Package	RISK	Risk Management (B-Plans)
WP1	Nrf24l01 didn't work properly	Because the nRF24L01 burned out, new ones had to be purchased.

Work Package	RISK	Risk Management (B-Plans)
WP1	Since the 22.4V battery was too heavy, the car could not turn and damaged the gears of the wheels.	a lighter and lower voltage battery was used, and the weight problem was solved. Also, the turning problem was eliminated.
WP1	Since OpenCV in Xavier is not compatible with CUDA, the object detection code was working very slowly, and we could not try another model because there was not enough memory.	I started using jetson nano again and object detection setups were made again and jetson nano was put back to the car again.
WP2	Because Jetson Nano was put back again, the camera's position and the weight of the car changed.	so, the speed and angle settings in the lane detection code were made again.
WP4	Yolov3 was a very complex model	Since fps is low with yolov3, a simpler model ssdmobilenet-v2 was chosen and fps was increased.

WORK TIME PLAN OF THE PROJECT:

A literature review was made from the very beginning of the project, and it was continued until the final test stages. The first work package took a lot of time. Because the nRF24L01module was burned in the first place, but I did not understand that it was burned, and I thought there was an error in the code. I tried to change the code constantly and then re-order nRF24L01. We gave it and learned that the error was caused by nRF24L01and it could be broken very easily. Later, I tried to move the car with the remote controller and adjust the speed parameters. Another problem was to power the jetson nano with a battery. I tried to power it with different batteries, but it did not work properly. When I read the Jetson nano documentation, it said that it could work with 5v and 4A, but even though I provided these values, jetson nano did not work. Later, when I talked to my advisor, was told that Jetson Xavier was in the lab and it would be more useful to use it, because jetson Xavier is faster than jetson nano and more useful for image processing.

There was no battery issue in Xavier, and I was able to run the lane detection code with the battery. However, since the 14.8V battery swelled, the 22.4V battery in the lab was given, but since this battery was too heavy, I had to balance the center of gravity. I printed a case on the battery and jetson Xavier with a 3d printer, but still the battery was too heavy for the car, and it was very difficult in turns. it was also damaging the gears of the wheel. so, I couldn't use that battery. A new 14.8v battery came to lab because someone finished their project and I was able to run the codes on the old system again, but I lost about 3 weeks because of this battery problem. Then I tried to run yolo v3 in jetson Xavier, but its fps was very bad. I tried v3-tiny, which is a different model of yolov3, but it was also very bad in terms of fps. The reason for the low fps is because OpenCV does not work with CUDA. I tried different installations to solve it, I tried to install OpenCV with sh files, then I tried to install it with SDK manager, but I could not succeed in any of them. I went back to jetson nano again and decided to set up all the requirements and run the car with the power supply because I couldn't use object detection in jetson Xavier in this process I lost 3 weeks as well. I installed object detection on nano and ran it together with lane detection, but since I lost a lot of time during this time, I could not collect data for the ai part, and we had to cancel that part.
I had to re-edit the lane detection code because the angle of the camera and the weight of the car changed because I switched to nano again. There was a delay in the work packages for 6-7 weeks due to the above problems. And some parts were not 100% successful.

Work package 1: Successfully building the car with all the modules included.
Completion Rate: 100%
Measure of Completion: the whole system of the car is prepared and the car moves smoothly

Work package 2: Detect the Lanes by the lane detection.
Completion Rate: 90%
Measure of Completion: Image results on the raspberry camera and the vehicle successfully completes the lap by following the lane

Work package 3: Avoiding obstacles by LIDAR.
Completion Rate: 100%
Measure of Completion: the lidar stops when the vehicle sees any obstacle in the road

Work package 4: Detection of pedestrian and stop sign with object detection.
Completion Rate: 100%
Measure of Completion: when a stop sign or a pedestrian is placed on the road, the car stops

Work package 5: Gathering footage.
Completion Rate: 50%
Measure of Completion: Collecting the dataset with the joystick controller and training the collected dataset

Work package 6 :Test the algorithms and make adjustments
Completion Rate: 90%
Measure of Completion: the car was tested on different roads and successfully completed the lap at a rate of 6/10

work package 5 is not 100% because the errors I mentioned above took a lot of time and I didn't have much time to collect data,

work package 6 is not 100% because I have no so much time that’s why I couldn't test it enough.

7 DEMO PLAN:

The demo of the project will give the chance to physically show the car I have built on the road I have drawn. The professors will choose the road they want with a modular and changeable road map, or they will ask for a trial on the road I have prepared. Later, when I leave the car on the road, it must complete the road by following the lanes. They can put obstacles so they can check if the lidar is working. Also, the professors will check whether the object detection is working by placing a stop sign or pedestrian on the road they want.

8 FINANCIAL EVALUATION:

Table 4: Financial Evaluation

Name	Unit Price	Quantity	Total	Description
Raspberry PI Camera Module	700 ₺	1	700 ₺	Cameras that we are going to use for the computer vision implementation.

L298N Motor Shield for Arduino UNO	97 ₺	1	100 ₺	To control the motors via controller we need a motor shield attached to the Arduino UNO board.
LIDAR Sensor	1900 ₺	1	1900 ₺	To be used for the distance measurement as well as mapping the environment to avoid crashes.
Jetson Nano Compute r	6000 ₺	1	6000 ₺	Computer that will be connected to the vehicle, the model is going to run on this compute r.
Motor & Wheels Module	25 ₺	4	100 ₺	Motorized wheels for the vehicle.
Arduino UNO	480 ₺	2	960 ₺	Will be the key distributer of the commands within the device.
Arduino UNO Joystick Shield	48 ₺	1	50 ₺	To use one of the Arduino UNO boards as a controlle r.
NRF24L01 WI-FI Module	25 ₺	2	50 ₺	To be connected to the controller and vehicle to act as a bridge of communication between these two devices.

7.4 V Battery	310 ₺	1	310 ₺	To provide the power source to the car.
Li-on Battery	100₺	2	200₺	To run the jetson nano
Mats	800₺	1	800	To design the road

10170₺ is planned but 2 lion batteries and 2 extra nRF24L01 were bought Also, extra materials were taken to design the road,800tl was spent for the extra road. so it cost 1050 Turkish liras more than planned.

9 RESULTS:

Building the car with all the modules included:

Both the connection between the motor driver and the Arduino, and the of the jetson nano with the Lidar and Arduino are shown in the figure below.

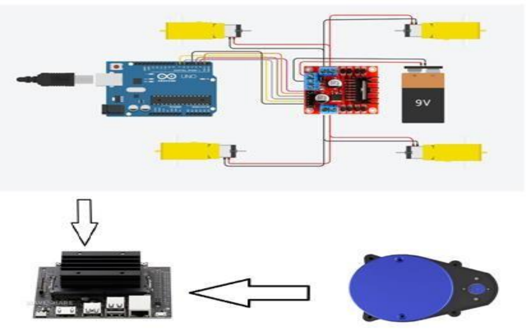

Figure-17: circuit diagram of the car

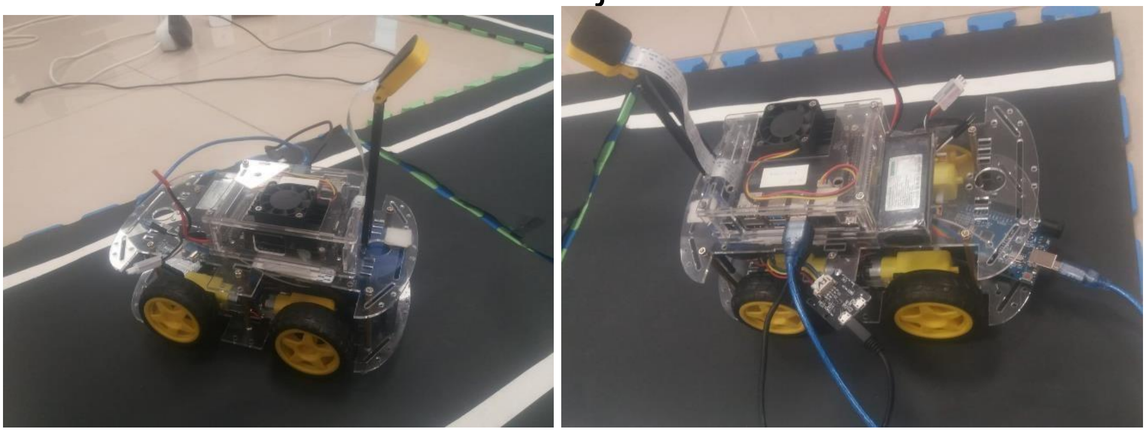

Figure:18-19 the view of the car when we put all the modules on the car

Detect the Lanes by the lane detection.

meangle: refers to the average of the left lanes.
meangler: refers to the average of the right lanes.
-1: indicates that there is no obstacle.

Figure-20: Turning Right Result

In the photo above, it is seen that the vehicle gives a right turn message when it sees a right turn.

Figure-21: Turning Left Result

Figure-22: Slight Right Result

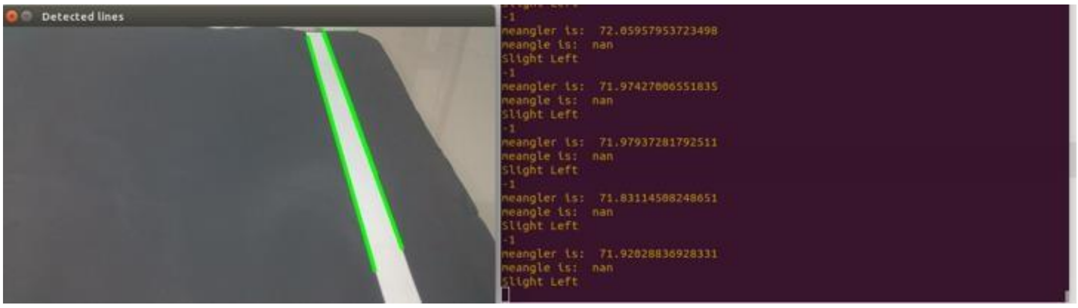

Figure-23: Slight Left Result

Figure-24: Go Forward Result

All the photos above show the decision made by the vehicle according to the frames. Looking at the terminal images, it can be said that the vehicle makes the right decisions. It makes these decisions according to the meangle and meangler variables. These variables represent the average slope of the right and left lanes. These decisions are also taken according to the model mentioned in the scope of the project.

Avoiding obstacles by LIDAR.

Figure-25: Obstacle Stop Result

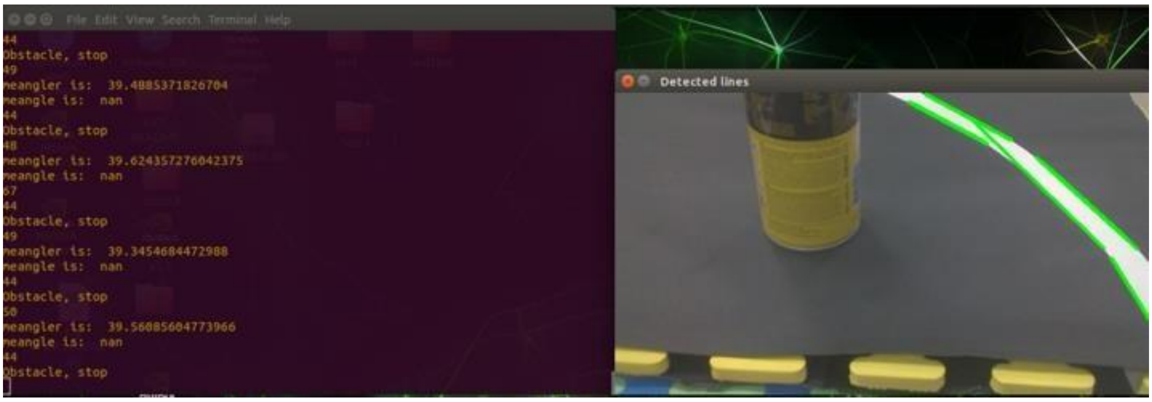

Figure-26: Obstacle Stop Result

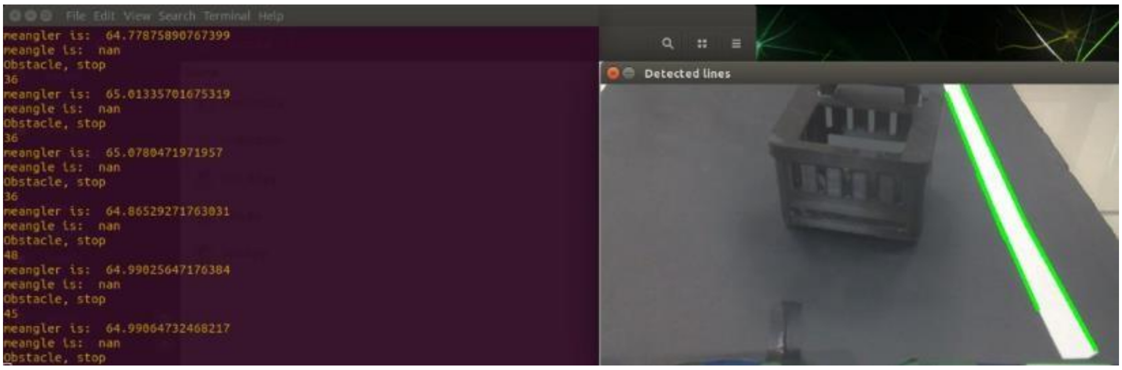

Figure-27: Obstacle Stop Result

The photos above show that, when an object is seen, the lidar notices it and makes the vehicle stop. When we look at the examples in the lane detection photos, it says -1 on the terminal, it shows that there is no obstacle on the road at that time, but in this terminal, it seems that the object is detected and makes the car stops.

Detection of pedestrians and stop sign with object detection.

Figure-28 Stop Sign Result

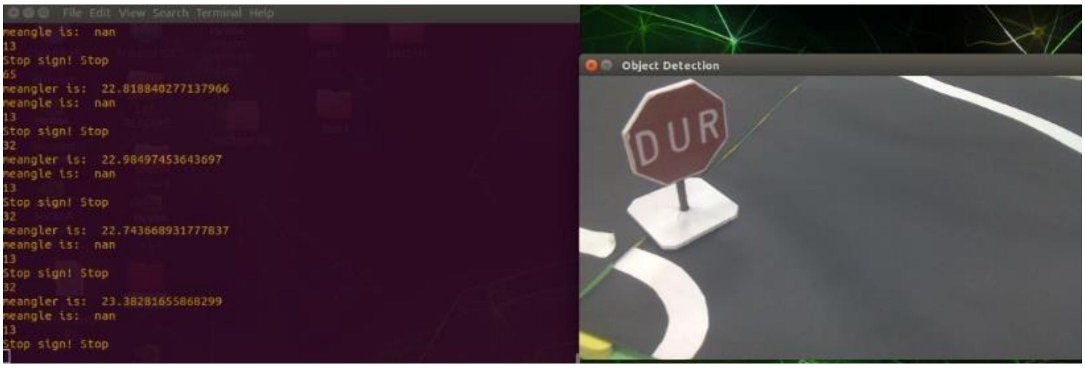

Figure-29 Stop Sign Result

The photos above show that when the camera sees the stop sign, the object is detected with ssdmobilenet-v2 and the car is ordered to stop. The text 13 on the terminal indicates that the classID of the stop sign .

10 DISCUSSION:

The objective of this project is to construct and develop an autonomous vehicle that showcases the potential of autonomous vehicles in a fundamental yet promising manner.
The main objective is for an autonomous vehicle to be built, which can act according to its environment without causing any issues from point A to B. Furthermore, it is aimed for this vehicle/car to have a compact design and form factor. If this car can be built with the expected functionalities in a tight package, the size can be changed according to the purpose of the vehicle.

At the beginning of the project, a lot of time was spent completing the design of the car. For example, I changed the code for weeks to establish the connections between the nRF24l01, but later I realized that the problem was not with the code, but with the nRF24l01, but I wasted a lot of time. Then I tried to solve the battery problem of Jetson Nano and tried to run it with different batteries, but without success. I started using Jetson Xavier because the battery problem could not be solved. Things were going well with him until the 14.8V battery in the lab broke. Since there was no other battery in the lab, I started using the heavy 22.4V battery and it was very difficult to move the car. Then another battery came, and I was able to use Xavier again, but we couldn't get enough fps in object detection because Xavier was not compatible with CUDA, so I decided to go back to jetson nano.

There are various technologies in the project. The first of these is lane detection. It is to ensure the movement of the vehicle on the road that I have drawn and to make a direction decision according to the lanes. Contrary to turning right and left, it also managed to go from the middle of the road while going on a straight road. That is, the tool will center the road even if it is started from the side of the road in the demo. Another feature is to scan the environment with YDLidar X4 and detect whether there are any obstacles on the road. After this scan, if an obstacle is detected, the vehicle is stopped. In the Object detection section, if a stop sign or pedestrian is seen on the road, the vehicle is stopped. Although it took a long time to come to the final stage of the car, all the features were successfully completed. On the first floor of the car, there is a motor driver, Arduino and lidar. On the upper floor, Jetson Nano, a Raspberry camera connected to Jetson Nano and behind it a 7.4V battery to power the motor driver was placed.

As seen in the circuit diagram of the car image in the conclusion, jetson nano controls the entire system. It transmits both the information it receives from the raspberry camera and the information it receives from the lidar to the Arduino. The motor driver gives power to the motors according to the speeds determined in the Arduino. Due to the delays I mentioned before, the work time plan 'was quite different from what was expected because a total of 6 weeks was spent to install OpenCV compatible with battery and Xavier with CUDA. This shows that not enough time is spent on other processes.

When we look at the conclusion part, we can say that most parts were completed, but the gathering footage part could not be completed due to the mishaps and delays. Even though the joystick controller part was completed, I could not complete this part because there was not enough time, but since the other features worked successfully, it can be said that about 90% of the project was completed successfully.

11 CONCLUSION:

The goal of this project is to develop a self-driving vehicle by combining advanced technologies. In this autonomous car, the vehicle uses images it receives to make decisions. These decisions are based on different factors. For example, the car detects lanes using a technique called Hough transform to ensure safe movement and turns, and it decides on the direction based on the detected lanes. To detect obstacles or objects on the road, the vehicle utilizes object detection and the YD Lidar X4 sensor. If an object is detected while scanning the environment with Lidar, the vehicle stops.

In the early stages of the project, a lot of time was spent on hardware, so there were a lot of deviations from the time in the work time plan. Looking at the results of the first work package, it is seen that the car is completely completed, and all its modules are fully and completely placed. There is a case for the Jetson nano and this case protects the jetson nano against impact. In addition, a 3D printer case was prepared for the Raspberry camera. Thanks to this cover, the camera is designed to look down and only see the road.

Looking at the results of the 2nd work package, as seen in the photos, the vehicle has successfully completed the turning operations. The vehicle can also center on the road, unlike only the right and left turns. In this way, if the car comes to on a lane, the vehicle will center the road.

These direction decisions were made according to the meangle and meangler parameters displayed in the terminal. Meangle and meangler represent the mean slopes of the left and right lanes. If both variables have a value of 45 or more, it indicates that they are on a straight road and the vehicle is in the middle of the road. If the meangle is Nan and the meangle is over 45 degrees, it means that the vehicle is in the right lane and the vehicle must center the road.

In the third work package, the vehicle was stopped with lidar. As seen in the results in the work package 3, the object was plotted and then the width of the object was calculated with the cosine theorem. To do this, the starting and ending angles of the object were determined and the result was calculated with the distance parameter. When the speed factor of the car is activated, if there is an object and its distance is less than 50 cm, the vehicle is stopped. Unlike Object detection, for Lidar it doesn't matter what the object is. If there is any obstacle on the road, the vehicle is stopped with lidar. In the conclusion part, it is shown with different examples.

For object detection, ssdmobilenet-v2 is used. The reason for using this model is that it is a simple model. The image settings of the camera are made as mentioned in 4.2. These values allow to reach the maximum fps. When only lane detection worked, 60fps was achieved, when only object detection worked, 25fps was achieved. However, when both were used together on the same camera, approximately 15-20fps was achieved.

work pack 5 could not be completed because there were many setbacks and work schedule delays. This part was only to be done to support the project but could not be completed, despite the lack of this work pack, the car still maintains its functionality and works.

As a result, 90% of the project has been successfully completed despite some setbacks.

12 FUTURE STUDIES:

Since the ai part could not be completed well, it is planned to continue this project. In this way, a decision mechanism will be formed according to both the lanes and the trained model, and the vehicle will be moved more accurately on straight roads and turns. Also, in lidar, only the stop is taking place now. Since the algorithm is ready, only speed settings and delays need to be adjusted.

It can also be made more compatible with real life by adding different traffic signs and some object's labels to object detection.

13 ASSESSMENT OF ENGINEERING COURSES:

It can be said that we have benefited from many of the lessons we have learned in this project. First, can be start with the circuits lesson. The benefit of this course is quite a lot, as the project includes hardware as well as software. Because, thanks to this course, we learned many subjects such as measuring voltage values, how electric currents are and how to calculate them, so when there is a problem in hardware, we can easily solve it. Thanks to the circuits we saw in the digital logic design course, we can build circuits on the Arduino much more easily. In addition, it can be said that the intro to CoE course is a great benefit. The car project we have done in the first grade also contributes to this project. We will also use python for the Raspberry camera. Thanks to the computer vision course we took this year, we will be able to do lane detection easily. The last and most important courses are machine learning and deep learning courses. Because we will make autonomous cars and there is no doubt that we will see the benefits of these lessons in the future.

14 REFERENCES:

[1] "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks" (G. Ren et al., 2017)

[2] "YOLO: You Only Look Once - Unified, Real-Time Object Detection" (J. Redmon et al., 2016)

[3] "SSD: Single Shot MultiBox Detector" (W. Liu et al., 2016)

[4] "Robust Lane Detection and Tracking in Challenging Scenarios" (Y. Wang et al, 2019)

[5] "Deep Learning-Based Lane and Road Marking Detection for Autonomous Driving" (M. Shen et al., 2018)
Systems

[6] "A Survey of LIDAR Technology and Its Use in Autonomous Vehicles" (M. V. S. Rao et al., 2019)

[7] "Lidar for Autonomous Vehicles: Recent Developments, Applications, and Outlook" (A. M. Patiño et al., 2020)

[8] "End-to-End Deep Learning for Self-Driving Cars" (M. Bojarski et al., 2016)

[9] "DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving" (C. Chen et al., 2015)

[10] "Learning to Drive from Simulation without Real World Labels" (P. Dosovitskiy et al., 2017)

[11] "Semantic Segmentation for Autonomous Vehicles: A Review" (K. Amende et al., 2020)

[12] "Failure Modes in Machine Learning for Autonomous Vehicles" (A. L. D. Moreira et al., 2019)

[13] "Legal and Ethical Aspects of Autonomous Driving" (A. K. Jain et al., 2018)

[14] "Impact of Autonomous Vehicles on Traffic Flow" (J. Jin et al., 2017)

[15] "Urban Planning for Autonomous Vehicles: Challenges and Opportunities" (S. Patel et al., 2018)

15 PROJECT ACTIVITIES AND WORK PLAN

Table 1 *The Work-Activity Plan for Project 1*

Work and Activity Project 1	**Responsible Group Member**	**Timeline**													
		1. week	2. week	3. week	4. week	5. week	6. week	7. week	8. week	9. week	10. week	11. week	12. week	13. week	14. week
1.Building the vehicle	Enes														
2.Testing the Vehicle's Functionalities	Enes														
3.Controlling the Vehicle via joystick	Enes														
4. Creating a Parkour for the Dataset	Enes														
5.Attaching the Camera Module	Enes														

Table 2 *The Work-Activity Plan for Project 2*

Istanbul Medipol University

School of Engineering and Natural Sciences

Graduation Project

Work and Activity Project 2	Responsible Group Member	Timeline													
		1. week	2. week	3. week	4. week	5. week	6. week	7. week	8. week	9. week	10. week	11. week	12. week	13. week	14. week
1. building the car with all the modules included:	Enes														
2. Detect the Lanes by the lane detection	Enes														
3. Avoiding obstacles by LIDAR	Enes														
4. Detection of pedestrian and stop sign with object detection.	Enes														
5. Gathering Footage	Enes														
6. Test the algorithms and make adjustments	Enes														

16.1 LIST OF WORK PACKAGES

***Table 2** Detailed Definition of Work and Activity*

WP No	Detailed Definition of Work and Activity
1	Preparing all parts of the car and completing the hardware part (nrRF24l01 connection, connection of motor driver, connection of jetson nano and arduino etc.)
2	Detecting lanes with hough transform and then determining which direction the vehicle will move in.
3	Scanning 360 degrees with lidar and ensuring that the car stops if any obstacle is detected on the road.
4	With object detection, detecting the pedestrian and stop sign and making the vehicle stop
5	driving the car with joystick controller for training the car and collecting the dataset.
6	After all the operations are done, testing with different roads and checking whether the car is working properly or not

***Table 3** Work package targets, their assessment, and the contribution of each work package to the overall project success.*

Work package	Target	Measurable outcome	Contribution to overall success(%)
1	Building the car with all the modules included:	the whole system of the car is prepared, and the car moves smoothly	30
2	Detect the Lanes by the lane detection	Image results on the raspberry camera and the vehicle successfully completes the lap by following the lane	25

3	Avoiding obstacles by LIDAR	the lidar stops when the vehicle sees any obstacle in the road	15
4	Detection of pedestrian and stop sign with object detection.	when a stop sign or a pedestrian is placed on the road, the car stops	15
5	Gathering Footage	Collecting the dataset with the joystick controller and training the collected dataset	10
6	Test the algorithms and make adjustments	the car was tested on different roads and successfully completed the lap at a rate of 6/10	5

Table 4 *The work package distribution to project team members: Who works on which work package? Specify the percentage contributions.*

WORK PACKAGE DISTRIBUTION						
Project Member	WP1	WP2	WP3	WP4	WP5	WP6
Enes Uğuz	100%	100%	100%	100%	100%	100%
Total	100%	100%	100%	100%	100%	100%

17. BUDGET

Table 5 *Proposed Budget in TL*

	ITEMS				
	PEOPLE	MACHINE-INSTRUMENT	MATERIALS	SERVICE	TRAVEL
IMU FUND	-	9370	800	-	-
SPONSOR COMPANY FUND	-	-	-	-	-
TOTAL	-	9370	800	-	-

Table 6 *Actual Budget in TL (what you spent indeed)*

	ITEMS				
	PEOPLE	MACHINE-INSTRUMENT *	MATERIALS*	SERVICE	TRAVEL
IMU FUND	-	10420	800	-	-
SPONSOR COMPANY FUND	-	-	-	-	-
TOTAL	-	10420	800	-	-

*Provide proforma invoice for machines and materials to be purchased.
*Provide technical specifications for machines and services to be purchased.
*Make a contract for services if necessary

Istanbul Medipol University
School of Engineering and Natural Sciences
Graduation Project

ENES UGUZ
STUDENT

EDUCATION
2018-2023 **COMPUTER ENGINEERING**

ISTANBUL MEDIPOL UNIVERSITY
GPA:3.28 5th place in department
100%Scolarship

2014-2018 **SCIENCE HIGH SCHOOL,**ENVAR COLLAGE
100%Scolarship

COURSES
- BECOMING AN ETHICAL HACKER COURSE(ATIL SAMANCIOGLU)
- WEB DEVELOPMENT COURSE (SADIK TURAN)
- BLOCKCHAİN, SOLİDİTY COURSE (FREE CODE CAMP)

EXPERIENCE

ANDASIS ELECTRONIC
Intern (August-September 2022)
Istanbul/Pendik

A web system was created using django, angular and mysql

ANDASIS ELECTRONIC
Intern (September-October 2022)
Istanbul/Pendik

network

SKILLS
PYTHON (+3 years)	●●●●
JAVA (+1 year)	●●
ADOBE PHOTOSHOP(+2 years)	●●●●
JAVASCRIPT (+1 year)	●●●
C++ (+1 year)	●●●

SUMMARY
Im Computer Engineering student at Istanbul Medipol University with CGPA 3.28 .I enroll my university with 100% scholarship and graduated as the 5th highest achiever in my program. I am proficient in Python, C++, and JavaScript, and I enjoy staying updated with the latest advancements in AI. I am eager to contribute to innovative projects and continue my career in the exciting world of artificial intelligence

PERSONALITY
-Creative
-Curious
-Hard working
-Practical Thinking
-Quick Learning
-Flexible

HOBBIES
- Playing guitar
- Fitness
- Ping pong

CONTACT
553 046 2533
enesuguz340@gmail.com

https://www.linkedin.com/in/enes-u%C4%9Fuz-987802216/

USKUDAR/ISTANBUL