REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

AUTONOMOUS FIRE DETECTION AND FIRST RESPONSE VEHICLE DEVELOPMENT WITH HOVERBOARD

20011629 — Barış BAKIM
20011025 — Anıl Kutay UÇAN

COMPUTER PROJECT

Advisor
Assist. Prof. Dr. Furkan ÇAKMAK

January, 2024

ACKNOWLEDGEMENTS

Yildiz Technical University, founded in 1911 in Istanbul, Turkey, is a prominent government institution with 10 faculties, 2 institutes, and approximately 25,000 students. It is also regarded as one of the greatest in the country.

We'd like to thank our instructor Furkan Çakmak for his significant assistance, continuous follow-up, and direction throughout our thesis on "Autonomous Fire Detection And First Response Vehicle Development With Hoverboard." His knowledge of 2D mapping, visual odometry, ROS and microprocessor programming was important in the project's success.

Barış BAKIM
Anıl Kutay UÇAN

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

CSS	Cascading Style Sheets
HTML	HyperText Markup Language
IP	Internet Protocol
JS	JavaScript
LAN	Local Area Network
ROS	Robotic Operating System
USB	Universal Serial Bus
UI	User Interface

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

AUTONOMOUS FIRE DETECTION AND FIRST RESPONSE VEHICLE DEVELOPMENT WITH HOVERBOARD

Barış BAKIM
Anıl Kutay UÇAN

Department of Computer Engineering
Computer Project

Advisor: Assist. Prof. Dr. Furkan ÇAKMAK

This project brings a new approach to robotics with using environmental monitoring, artificial intelligence, and machine learning. A computer is integrated into a mobile robot, empowering it to autonomously navigate and survey a room, continuously creating closed area maps and capturing heat values. The project adds new features like machine learning algorithms specifically designed for fire detection during the robot's autonomous operations.

With the help of sensors, including lidar and heat sensor, the robot identifies temperature anomalies and uses machine learning for precise fire detection. The collected data, mapping, temperature distributions, and machine learning informed fire detection results, is transmitted to a central server accessible over the local area network. This facilitates real-time, global monitoring of the environment.

An important feature of the robot is its proactive response mechanism to fire. It automatically sends alert to user interface.

Keywords: Autonomous Robotics, Environmental Monitoring, Machine Learning for Fire Detection, Data Transmission, Intelligent Emergency Response

ÖZET

HOVERBOARD İLE OTONOM YANGIN TESPİTİ VE İLK MÜDAHALE ARACI GELİŞTİRME

Barış BAKIM
Anıl Kutay UÇAN

Bilgisayar Mühendisliği Bölümü
Bilgisayar Projesi

Danışman: Dr. Ögr. Üyesi Furkan ÇAKMAK

Bu proje, çevresel izleme, yapay zeka ve makine öğreniminin kullanımıyla robotiğe yeni bir yaklaşım getiriyor. Mobil bir robota entegre edilen bir bilgisayar, robotun otonom olarak gezinmesine ve bir odayı incelemesine, sürekli olarak kapalı alan haritaları oluşturmasına ve ısı değerlerini yakalamasına olanak tanıyor. Proje, robotun otonom operasyonları sırasında yangın algılaması için özel olarak tasarlanmış makine öğrenme algoritmaları gibi yeni özellikler ekliyor.

Lidar ve ısı sensörü de dahil olmak üzere sensörlerin yardımıyla robot, sıcaklık anormalliklerini tespit ediyor ve hassas yangın tespiti için makine öğrenimini kullanıyor. Toplanan veriler, haritalama, sıcaklık dağılımları ve makine öğrenimi bilgili yangın algılama sonuçları, yerel alan ağı üzerinden erişilebilen merkezi bir sunucuya iletilir. Bu, çevrenin gerçek zamanlı, küresel olarak izlenmesini kolaylaştırır.

Robotun önemli bir özelliği yangına proaktif tepki mekanizmasıdır. Kullanıcı arayüzüne otomatik olarak uyarı gönderir.

Anahtar Kelimeler: Otonom Robotik, Çevresel İzleme, Yangın Algılama için Makine Öğrenimi, Veri İletimi, Akıllı Acil Durum Müdahalesi

1
Introduction

This project aims to provide an effective solution to the fires occurring in industrial areas and the resulting loss of life and property. The overheating of intensive and continuously operating machines can lead to serious damage and fires within their structures. Additionally, fires coming from external factors in workplaces have significant risks [1]. In this project, an hoverboard based robot is used to map the workplace's temperature, perform SLAM, and autonomously detect temperature anomalies and fire during this process, triggering alarms. Furthermore, autonomous fire detection and intervention are made by using lidar, temperature sensors and fire sensor.

The robot used in the project is constructed on a specially designed chassis placed on a hoverboard. The robot's hardware is enhanced with temperature sensors, sonar sensors, fire sensor and lidar. Using the SLAM algorithm, the robot creates 2D maps and temperature maps. These maps will serve as fundamental data sources for the robot to autonomously detect temperature anomalies. Additionally, an AI-based autonomous fire detection system is integrated, and the data obtained by the robot is displayed via a control interface accessible over the internet.

One notable feature of the project is the design of an robot capable of autonomously controlling the temperature of the entire workplace, not just specific machines. This approach aims to reduce the risk of fire in machines without sensor placement and throughout the overall workplace. Innovative elements in the project include technologies such as heat map creation, designing an internet-accessible control interface, developing an AI-based fire detection and intervention system, and the use of a recycling-focused hoverboard.

2
Preliminary Examination

The project aims to address industrial fires and associated losses by developing an robot capable of mapping temperature in workplaces, creating occupancy grid map by using SLAM algorithm, detecting temperature anomalies, detecting fire from camera and intervening in case of a fire. The robot, built on a recycled Hoverboard, integrates temperature sensors, sonar sensors, and an Lidar sensor. The innovative aspects include autonomous temperature mapping, AI-based fire detection, and intervention, an accessible control interface over the internet, and the use of a recycled Hoverboard. The method involves designing the robot, integrating sensors, employing SLAM algorithms, AI-based fire detection, developing an internet-accessible control interface, and real-world testing. The project's significance lies in enhancing industrial fire safety by minimizing damages through comprehensive temperature control. Moreover, in [2] work, the deep learning approach for fire detection through camera is observed to have great performance in real life fire detection.

2.1 Literature Review

The project has 3 main part: SLAM, navigation and fire detection. Because most of the projects doesn't completely match these criteria a general comparison table is created. Only the techniques that are used in the article are written.

[3] The research is published in 2015. It is mainly focused on detecting fire. Therefore, it uses 3 different sensors and a sensor fusion algorithm to achieve this. The designed robot is able to move and avoid obstacles thanks to its ultrasonic sensors. However, it doesn't map the environment and localize itself. The sensors that are used for fire detection are smoke sensor, temperature sensor and fire sensor. Those sensors and the algorithm successfully detects fire.

[4] The research is published in 2023. It is a design for well equipped robot that is able to do SLAM, autonomously detect fire and fight against it with fire extinguishers.

Instead of water, the CO extinguishers were chosen for the project. Lidar sensor, IMU and encoders are used for Cartographer SLAM algorithm to map the environment and localize the robot. Infrared thermal camera is combined with YOLOV4 deep learning algorithm to detect fire. The designed robot worked successfully. It was able to map the environment, detect fire and respond to it.

[5] The research is published in 2015. It is a LEGO based robot design aiming to detect fire. In the project color sensor, temperature sensor and light sensor are utilized to detect fire. Those sensors are fused with type-2 fuzzy logic. The system is simulated in Matlab. The robot also have an ultrasonic sensor which would help to avoid obstacles. However, it isn't enough to map the environment and localize itself.

[6] The research is published in 2020. It is a robot design that would sense the fire. It doesn't have any sensor that would measure distance so it isn't possible to autonomously search the area. The robot has flame sensors for fire detection. In the research the effectiveness of the fire sensor is examined and the following conclusion is made: the fire sensor is useful when there isn't lots of light. Therefor, this system might only effectively work in dark places.

ID:	A	B	C	D
ARTICLE:	Development of mobile robot with Sensor Fusion Fire Detection Unit	An Indoor Autonomous Inspection and Firefighting Robot Based on SLAM and Flame Image Recognition	Fire Detection Robot using Type-2 Fuzzy Logic Sensor Fusion	Autonomous Safety Mechanism for Building: Fire Fighter Robot with Localized Fire Extinguisher
DATE:	2018	2023	2015	2020
AUTHOR:	Hilmi Saygın SUCUOĞLU	Sen Li, Yunying Li, Chunyong Feng, Yijin Gao, Jialuo Yang, Guangchao Sun, Dan Zhang	Xuqing Le	Izyan Yahaya
Mapping:	-	Cartographer SLAM	a type-2 fuzzy logic system	-
Navigation:	Fuzzy logic	SLAM	type-2 fuzzy logic applications	-
Fire Detection:	Fuzzy logic	YOLOv4 deep learning	T1 FLS and T2 FLS	a supervisory control algorithm

Figure 2.1 Comparison Between Current Studies

A: [3], B: [4], C: [5], D: [6]

3
System Analysis and Feasibility

The project aims to map the temperature distribution within workplaces, implement autonomous anomaly detection based on the temperature map, and autonomous fire detection from camera. This comprehensive strategy ensures a proactive and effective approach to enhancing workplace safety by identifying potential hazards and mitigating risks associated with overheating machinery.

3.1 Feasibility

3.1.1 Time Feasibility

A three-month period was determined to be proper for project completion. The Waterfall approach was used to create the time feasibility aspect.

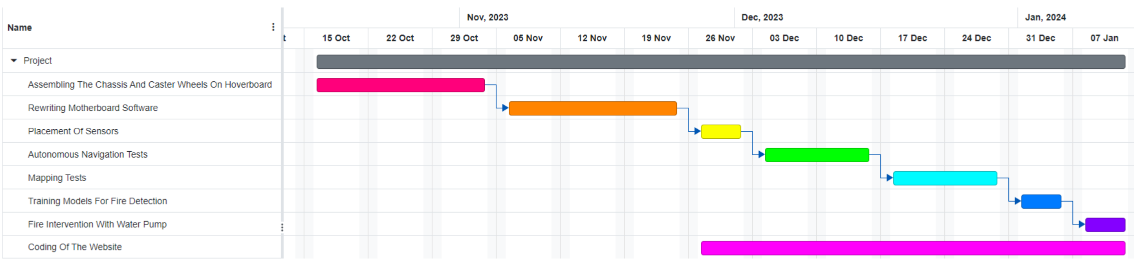

Figure 3.1 Gantt Schema for Project

3.1.2 Technical Feasibility

The technical feasibility of the proposed robot design with a differential drive system and integrated sensors for navigation, obstacle avoidance, and fire detection appears promising. The utilization of a hoverboard chassis for the robot, coupled with a robust set of sensors including 2-D Lidar, temperature sensors, a camera, and sonar sensors, enhances its perceptual capabilities. The incorporation of an open-source firmware and the implementation of an Extended Kalman Filter (EKF) for sensor data synchronization demonstrate a thoughtful approach to hardware and software integration. The use of ROS2 as an interface for seamless communication between sensors and processes ensures a structured framework. The application of artificial intelligence for fire detection, along with the development of a web-based user interface, adds significant functionality to the system. The planned simulation tests in Gazebo before real-world testing further strengthen the technical viability of this innovative robot design, emphasizing its potential for effective environmental monitoring and intervention.

3.1.3 Legal Feasibility

The objective of our project is to implement an autonomous fire mitigation system utilizing robotics and artificial intelligence. Ethical concerns and potential harm are negligible, eliminating the necessity for ethics committee approval. Open-source software forms the basis of our system, eliminating concerns related to software licensing violations. The development process is internal, guaranteeing adherence to licensing requirements.

3.1.4 Economic Feasibility

The cost of the hardware used is 40000 TL, since there are two computers with each one costing 20000. Hoverboard costs 6000 TL, Lidar costs 2000 TL, Chassis And Caster Wheels costs 700 TL, Arduino Mega 500, Heat Sensor 50, RGB camera 800, Fire Sensor 50, An employee works for 40000 TL monthly. Two employees who work quarter-time for 4 days a week, paid 2 * 40.000 * 0.8 * 0.25 = 16000. The payment for two individuals for 3 months is 16000 * 3 = 48000. So for this project total salary is calculated as 48000 TL. The total budget is 40000 + 6000 + 2000 + 700 + 500 + 50 + 800 + 50 + 48000 = 98100 TL

Table 3.1 Components and Prices for the Project

Component Name	**Price (TL)**
Computers (2 units)	20000
Hoverboard	6000
Lidar	2000
Sonar Sensor	100
Chassis and Caster Wheels	700
Arduino Mega	500
Heat Sensor	50
RGB Camera	800
Fire Sensor	50
Employee Salary (3 months)	48000
Total Budget	**98100**

3.2 Elements of the System

The goal of this project is to create a robot that finds temperature anomalies to prevent overhearing and give a first response to the fire. By doing so, decrease need of hardware, use of human use of human and data resources from heat sensors. The Table 3.2 below provides an overview of the project's two resources which are human and data resources.

Table 3.2 Resources which are used for the project

Resource	**Element**
Human Resources	Barış Bakım, Anıl Kutay Uçan
Data Resources	Heat Sensor, Lidar, Camera, Sonar Sensor, Fire Sensor

3.3 Use Case Scenarios and Diagrams

Use case scenarios and diagrams are used to evaluate the possible outcomes and risks of the project.

Use Case Title: Create Heat Map
Primary Actor: User
Supporting Actor: Robot
Pre-conditions: The user started the robot and SLAM algorithm is started.

Main Success Scenario:

1. The Robot gets information from Lidar.
2. The Robot feeds the SLAM algorithm with the sensor data.
3. SLAM algorithm uses the data to update the map and localize the robot.
4. The Robot gets data from temperature sensor.
5. The Robot searches for the odometry data for the time which the temperature data is received.
6. The Robot adds the temperature data to the found location in the map.

Extensions:

- 1a. The Robot can't get information from Lidar.
  - 1a1. The Robot gives an error

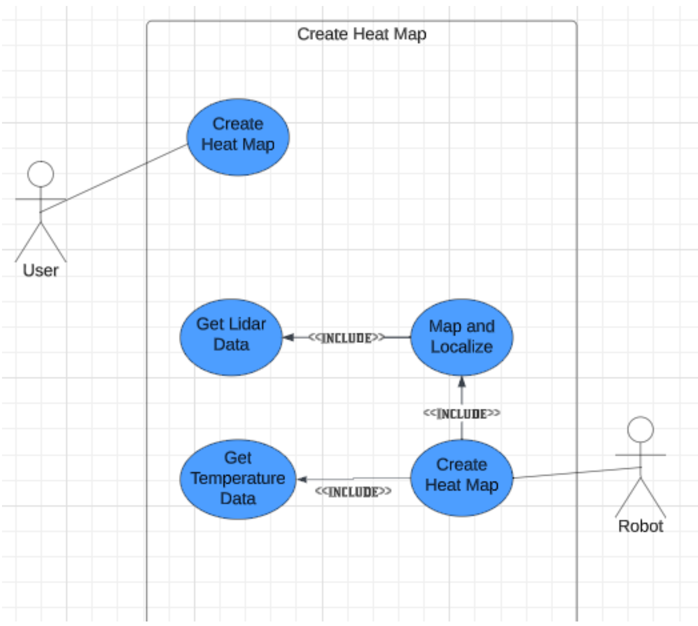

Figure 3.2 Use Case Diagram For Create Heat Map

Use Case Title: Fire Detection With Heat Map
Primary Actor: Robot
Pre-conditions: The heat map is generated

Main Success Scenario:

1. The robot gets data from temperature sensors
2. The robot compares the data with the heat map value
3. The received data is significantly higher than the heat map value
4. The robot sends alarm

Extensions:

- 3a. The received data is close to the heat map value
  - 3a1. The Robot continues traveling

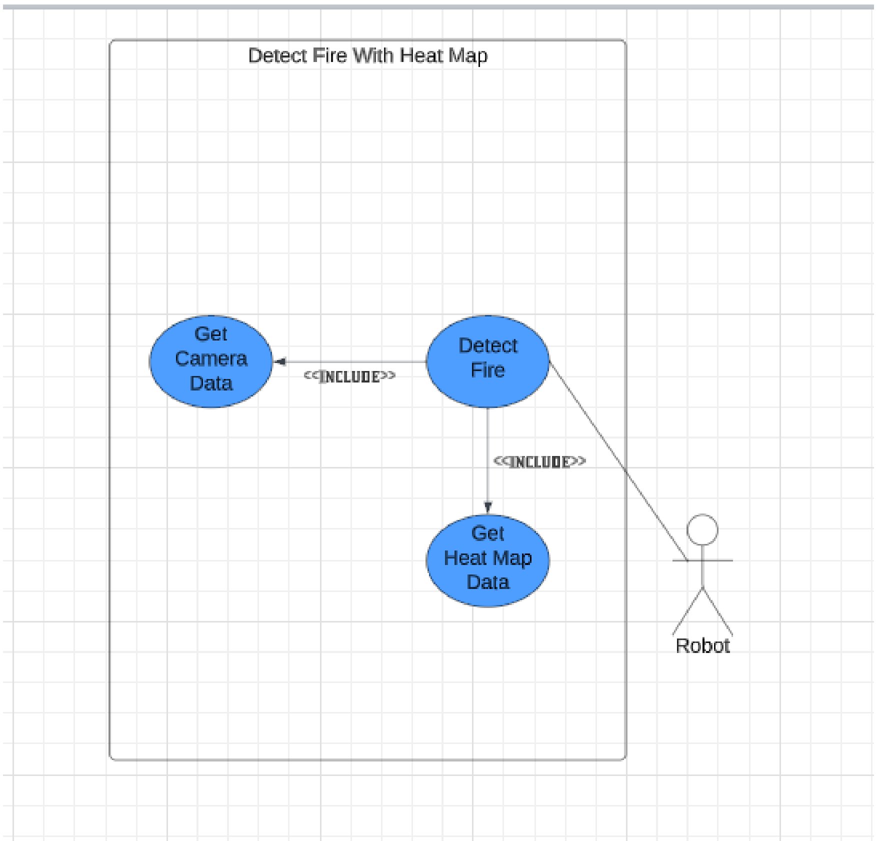

Figure 3.3 Use Case Diagram For Detect Fire With Heat Map

Use Case Title: Fire Detection With Camera
Primary Actor: Robot
Pre-conditions: Camera sensor is ready to use

Main Success Scenario:

1. The robot gets data from fire sensor
2. The fire sensor detects fire
3. The robot gets data from camera sensor
4. The robot gives the data to pre-trained fire model
5. The model detects fire
6. The robot sends alarm
7. The robot starts its auto aiming system
8. The robot throws water towards the fire

Extensions:

- 2a. The model doesn't detect fire
  - 3a1. The Robot continues traveling
- 5a. The model doesn't detect fire
  - 5a1. The Robot continues traveling

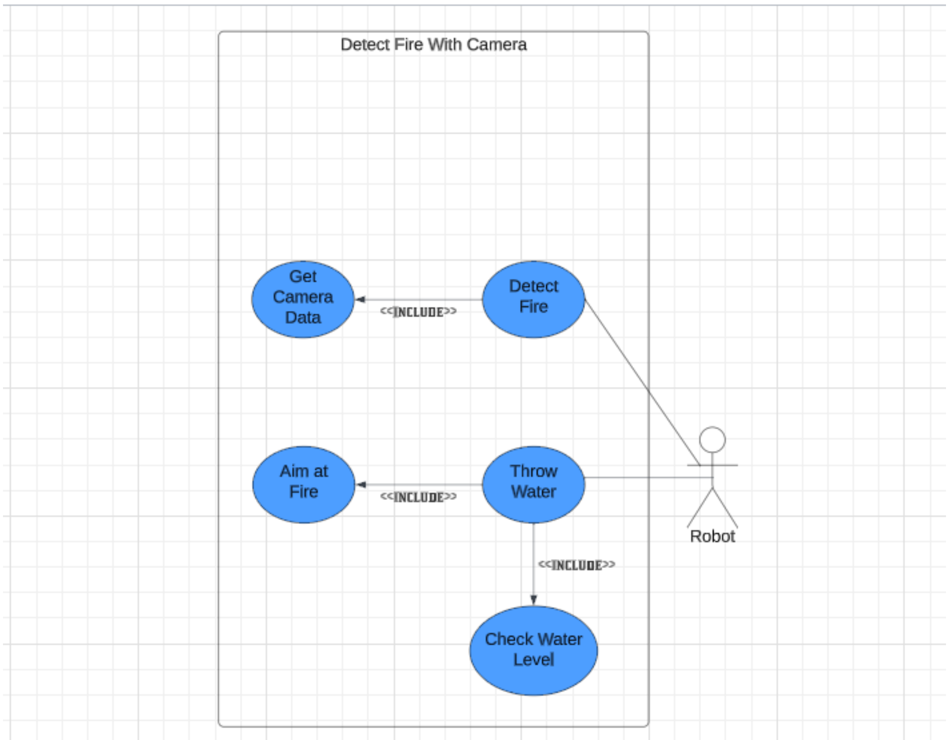

Figure 3.4 Use Case Diagram For Fire Detection With Camera

4
System Design

The goal of this project is to develop an robot that detects fire or overheated systems that would turn into fires and send alarms to prevent any damage.

4.1 Hardware Design

In this research, it was decided to fabricate a robot with a differential drive system, comprising two fixed wheels and one caster wheel that is placed on an uniquely designed hoverboard chassis (Figure 4.1). This choice underscores the significance of recycling. The technical specifications for the envisaged robot are as follows: weight: 12 kg, maximum payload: 100 kg, motor power: 350W, maximum speed: 20 km/h. To enhance the robot's perceptual capabilities for robust navigation, obstacle avoidance, and environmental interaction sensors are utilized. The chosen sensors include 2-D Lidar for precise distance measurements, temperature sensors for environmental monitoring, sonar sensors for enhancing safety, fire sensor for fire detection enhancement and a camera for visual data interpretation. The integration of sensors into the robot serves as a critical augmentation to its functionality, enabling a more nuanced and adaptive response to its surroundings. Sonar, fire and temperature sensors are connected to Arduino which communicates with the onboard computer. The 2-D lidar sensor is directly connected to the computer. The following sections delineate the rationale behind the selection of each sensor type and elucidate the potential contributions to the robot's overall performance.

1. 2-D Lidar Sensor: The incorporation of Lidar sensors emerges from the imperative for accurate distance measurement and environmental mapping. Lidar provides high-resolution, two-dimensional distance data, facilitating precise localization and mapping (SLAM) capabilities. This is paramount for the robot's navigation in dynamic environments, ensuring real-time awareness of its surroundings and the ability to navigate through complex terrains with optimal efficiency.

2. Temperature Sensors: The integration of temperature sensors addresses the exigency to monitor environmental conditions during the robot's operation. The temperature data collected by the sensors will be utilized to create a heat map by combining it with the 2D map created by Lidar. Moreover, variations in temperature can be a warning against an overheated system that might catch fire. Therefore, the sensors would catch the anomalies in the temperature by comparing it with the created heat map and provide an early fire detection system.

3.Fire Sensor: The addition of the fire sensor improved the accuracy of the fire detection which only uses camera. Even though camera can detect the fire through images, it also has false positives which is a big problem for the industrial utilization. The fire sensor detects the infrared light which is emitted by fire. Therefore, the fire sensor is integrated to decrease the number of false positives and improve the accuracy of the algorithm. Even though the camera detects a fire it first waits for the fire sensor to detect infrared light. If the sensor doesn't detect any infrared light, this result gets evaluated as false positive.

4. Camera and Image Processing: The inclusion of a camera, coupled with advanced image processing techniques and a fire sensor, facilitates visual data interpretation. When combined with AI, this sensor augments the robot's perceptual prowess to detect fire.

5. Sonar Sensors: The incorporation of sonar sensors are to enhance the safety and situational awareness. the sensors strategically placed around its perimeter to detect nearby objects. The primary objective is to enable the robot to autonomously stop in the face of unforeseen obstacles, thereby minimizing the risk of damage to objects and potential harm to individuals in its vicinity.

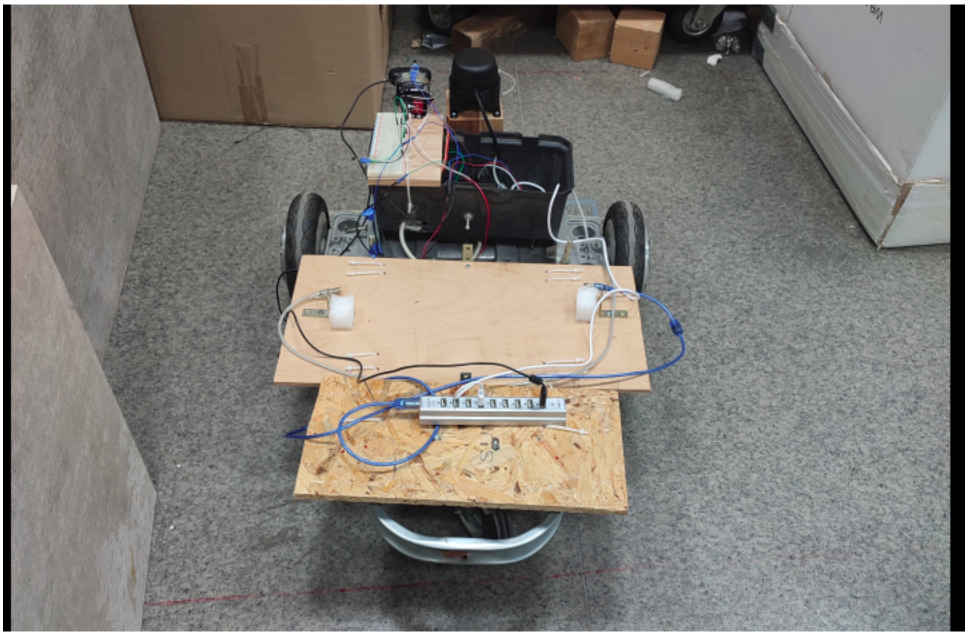

Figure 4.1 Designed Hoverboard Chassis

4.2 Software Design

4.2.1 Hoverboard Firmware Update and Communication

The project involves the modification of the firmware embedded in the hoverboard's onboard microcontroller to enable controlled maneuvers and real-time monitoring. The chosen firmware, released under the GPL v3.0 license by [7] in 2019, represents an open-source solution. The implementation of this firmware allows for serial communication with the hoverboard's microcontroller, facilitating the transmission of movement commands and retrieval of crucial status information [7]. To communicate with the hoverboard's motherboard, a microcontroller is used which works as a bridge between the computer and the motherboard. To communicate with the microcontroller from the computer, a message format is created. The format utilizes 2 bytes for data transmission. The high end nibble of the high byte specifies the message type. The other parts specifies the message.

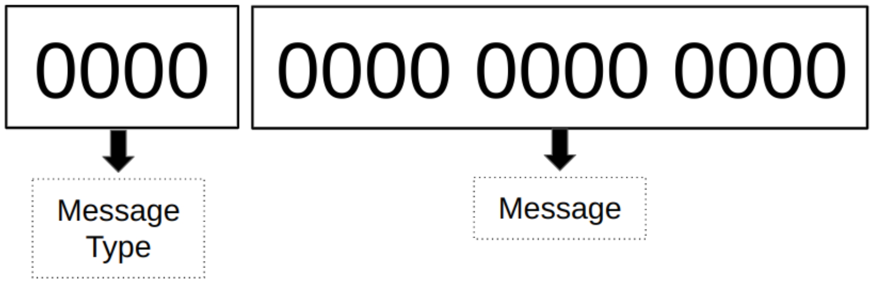

Figure 4.2 Message Format In Bytes

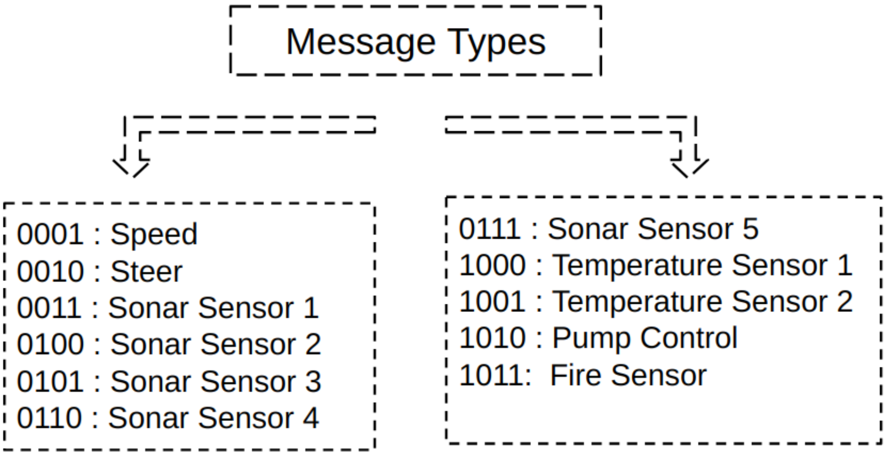

Figure 4.3 Message Types

For the communication protocol, serial communication is used with the baud rate of 115200. The sensors are connected to the microcontroller from the designated pins. The developed software is capable of sending the data of 5 sonar sensors, 2 temperature sensor, 1 fire sensor from the microcontroller to the computer. Moreover, the software is able to receive speed, steer and start pump information. It converts the speed and steer information into a message type that the motherboard can understand and sends this message with the serial communication protocol to the motherboard from another serial port. Moreover, it evaluates the start pump message to start the pump by giving voltage from the specified pin which goes to the base of the transistor.

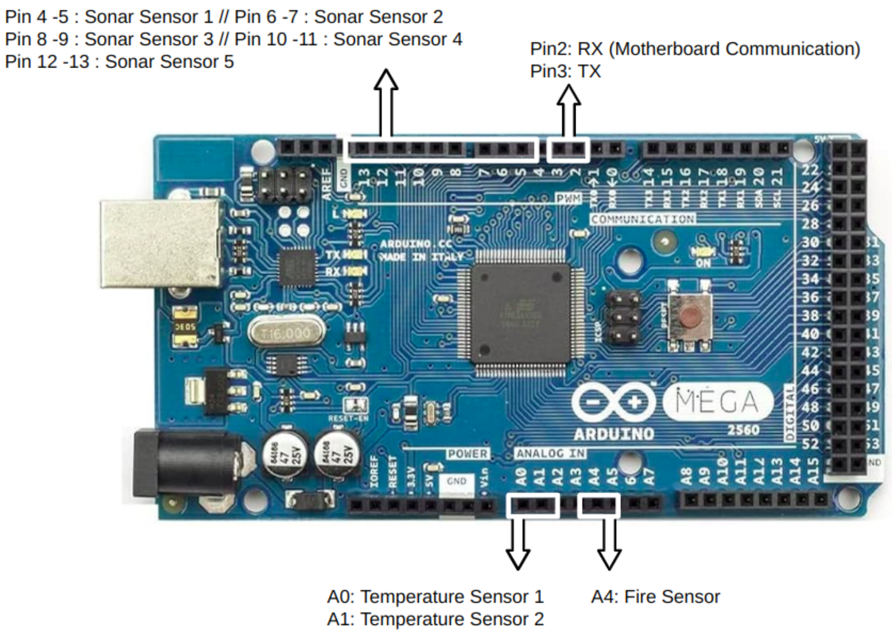

Figure 4.4 Arduino Connections

4.2.2 Fire Detection

An AI trained model by [8] is used to detect fire from an image. In order to improve the accuracy and prevent any false positives, an infrared fire sensor is added. The sensor is placed on top of the camera. It is able to detect infrared light which is emitted from the fire up to one meter. To detect fires that are far away the sensor quality should be changed. However for the development phase of the project the utilized sensor is decided to be enough. If the sensor detects fire, the image data gets evaluated by the model. If the model also detects fire, it publishes an warning message from a topic and the image data with the detected fire is encapsulated in a blue box. In brief, when the fire detection sensor works and detects fire, it warns the system to start image processing to find the fire's exact position according to the camera. Then node publishes a warning message and the processed image.

4.2.3 Heat Map Creation

The heat map creation software is created to store temperature data of an area which will be used to detect temperature anomalies that can be turned into fire. The software has 5 main features. It is able to create the heat map of the environment, reset the created heat map, save the heat map in a png file, load the heat map from the saved png file and detect anomalies in the heat map. To use this software it is necessary to publish odometry and temperature data.

4.2.3.1 Mapping Mode

In the mapping mode, the heat map creation software gets the temperature and odometry data. The odometry data is received from the transform library which calculates the transform from the odom to the base link frame. The first map is created statically and the message type of the map is occupancy grid map. It is possible to set the origin, resolution, width and height of the map. At first the created map is filled with 0s which means that the temperature data isn't set yet. When the node receives the temperature data, it gets the current transform from odom to the base_link. It calculates which point in the map the robot received the temperature data and it changes the value of that index with the temperature data. The software is also able to update the temperature value of the nearby positions if specified. It is able to reset the map by subscribing a reset topic. When the reset command is received, it dumps the current heat map and creates a new one. In both modes, it is able to save and load maps from a given file. To save the heat map, it uses png file format. It first finds the equivalent color for the temperature value for the user to be able to interpret the image. Then it updates the color value of the pixel with the calculated value. After that, it saves the created image into the given file path. It also creates a yaml file that specifies the map's properties. To load the map from the png file, it gets the path of the yaml file and then reads the png file and converts it into an occupancy grid message. The created message can be used to map the environment again or detect temperature anomalies.

4.2.3.2 Detection Mode

In the detection mode, when it receives the temperature data, it gets the position of the robot. Then it uses the received odometry to find the map index of the position. It compares the temperature value gotten from the map with the temperature value received from the sensor. If it is above the specified threshold, it creates a warning and publishes it. This is one of the key component of the project. It is possible for the fire to be invisible or not started yet. The fire wouldn't start till a certain temperature so with this software, it is possible to detect the temperature anomaly and prevent the fire before it even takes place. In the detection mode, if there isn't any temperature value for the current robot's position, the position is marked with the received temperature value in the map. Then this value is utilized to detect any anomalies in that position.

4.2.4 Synchronization and Hoverboard Control

To achieve accurate temporal synchronization between the sensors, odometry, camera, and temperature data must be generated with concurrent timestamp. Given that the algorithms using these data operate as separate processes, a structured communication framework is essential. For this purpose, ROS2 (Robotic Operating System) is employed, serving as an interface that processes data from the robot's sensors and facilitates command transmission [9]. Additionally, ROS2 is utilized to establish a software framework, leveraging its peer-to-peer communication system for seamless inter-process communication. The message template in ROS2 includes timestamp information, allowing for temporal synchronization of data across processes. Leveraging this timestamp information, a more consistent temperature is generated by utilizing odometry data from the same time instance. Simultaneously, a 2D obstacle map and a temperature map is generated to provide a comprehensive understanding of the robot's environment.

An hoverboard control node is written for the onboard computer in python. ROS2 is used for the communication between processes. The hoverboard controller node is subscribed to cmdVel topic which is the topic used for velocity information. The received velocity message needs to be converted into a message that the hoverboard can understand. For the transition process the rpm values for the given velocity commands are observed and the following formula is developed to convert from m/s to hoverboard speed.

$$\text{RPM} = \frac{\text{speed} \times 60}{\text{TIRE\_CIRCUMFERENCE}}$$

$$\text{Hoverboard Speed Message} = 100 + \left(\frac{\text{RPM - 40}}{6} \times 10\right)$$

4.2.5 Web Based User Interface

An internet-accessible interface will be designed, with a dedicated computer on the robot serving as a server. Apache server, along with a servlet container like Flask or Tomcat, will handle incoming POST and GET requests, directing them to the appropriate processes. The server's primary functions include relaying the temperature map, robot location, sensor readings, and temperature anomalies to the user. Additionally, the server will transmit control commands received from the user to the robot. The web interface will be developed using HTML, CSS, and JS, while algorithmic implementations will use Python. Simulation tests will be conducted in Gazebo, with the system tested in a simulated environment after constructing the robot's urdf model. Successful simulation results will pave the way for real-world testing.

5
Experimental Results

The experiments are conducted in simulation environment and real life. The developed software are first tested on simulation environment and then integrated into the real life application.

5.1 Simulation Environment

Simulation plays a crucial role in robotics development, allowing engineers and researchers to test and refine their algorithms without the constraints and risks associated with physical prototypes. In the world of robotics simulation, Gazebo stands out as a popular tool, providing a realistic environment for simulating robot behaviors. To test the algorithms in the simulation environment, a robot model and a world is created. The robot model is a differential wheeled robot that has 2 wheels and 1 caster wheel. It has a lidar sensor on top of it. The world consist of 4 walls and inside of the walls there are couple of obstacles.

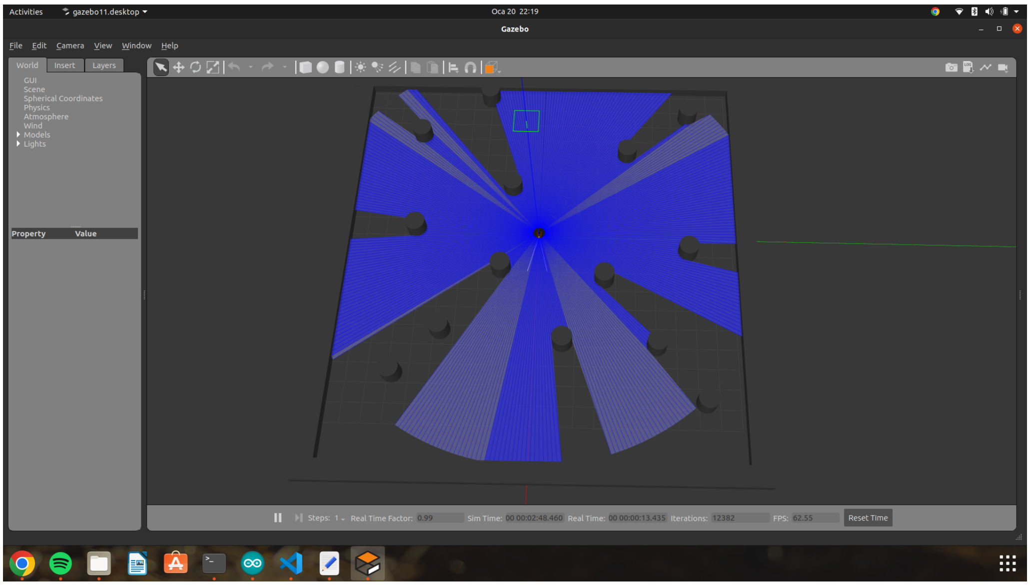

Figure 5.1 Gazebo Simulation Environment With Lidar

5.1.1 Testing SLAM and Heat Map Algorithm in Gazebo

To create the map of the environment slam gmapping and laser scan matcher is utilized. Laser scan matcher creates an odometry topic and a transform information from odom to base_link by using laser scan matching with ICP algorithm. By using this data it is possible to calculate where the robot is relative to the starting place. Moreover, Slam Gmapping gets the odometry and lidar information and creates a map of the environment in occupancy grid map format. For mapping algorithms Gmapping and Slam Toolbox are tested. They both performed similar and was able to map the area without loosing odometry. However, it is observed that slam toolbox isn't suitable for dynamic environments because it doesn't remove the created obstacles from the map. This is a grave issue when working in a dynamic environment such as fabrics. Therefore, it is decided to use gmapping.

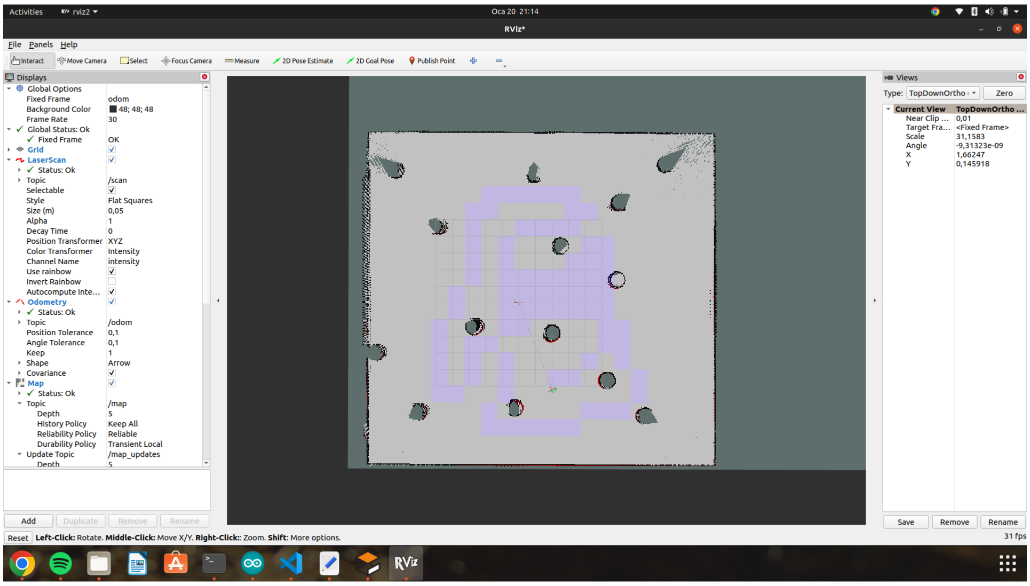

Figure 5.2 Gmapping And Laser Scan Matcher Algortihms Test

The heat map creation algorithm is tested simultaneously with gmapping and laser scan matcher algorithms. Because it needs temperature data, a test software is written that simulates a temperature sensor. The heat map creation algorithm was able to map the received temperature data with the correct position. In mapping mode, it successfully updates the map with the received temperature data. In the detection mode, it was able to detect the temperature anomalies and publish a warning message that has the coordinates of the anomaly. Moreover, in detection mode it was able to update the heat map if there is no temperature value for the given position. The heat map save feature is also tested to save the heat map in a png format. For the image file, it successfully converts the temperature value to color and saves the heat map into the given path. Moreover, the software is able to read the png file and converts it back to occupancy grid map format and performs with full functionality.

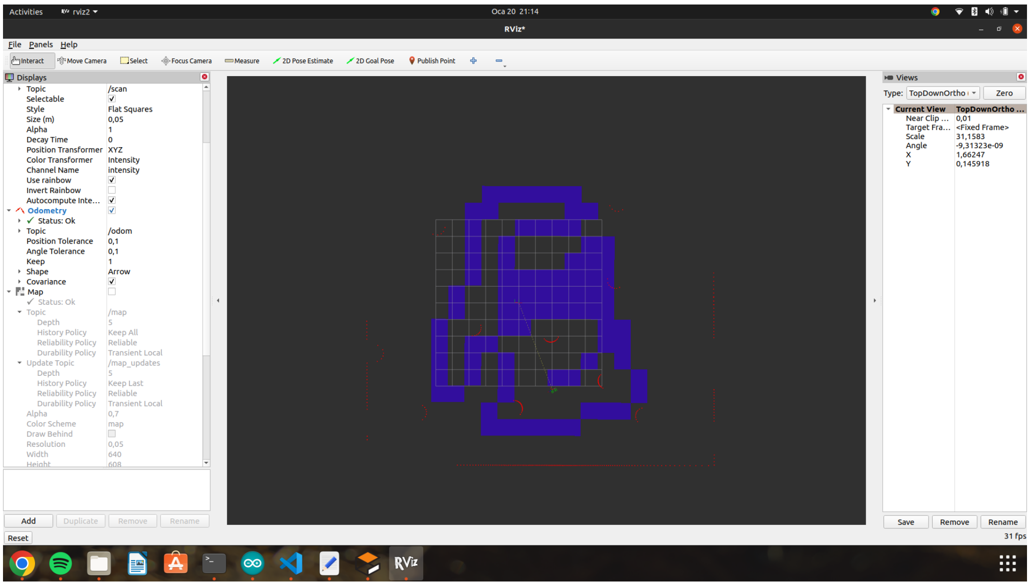

Figure 5.3 Heat Map Visualization in Rviz

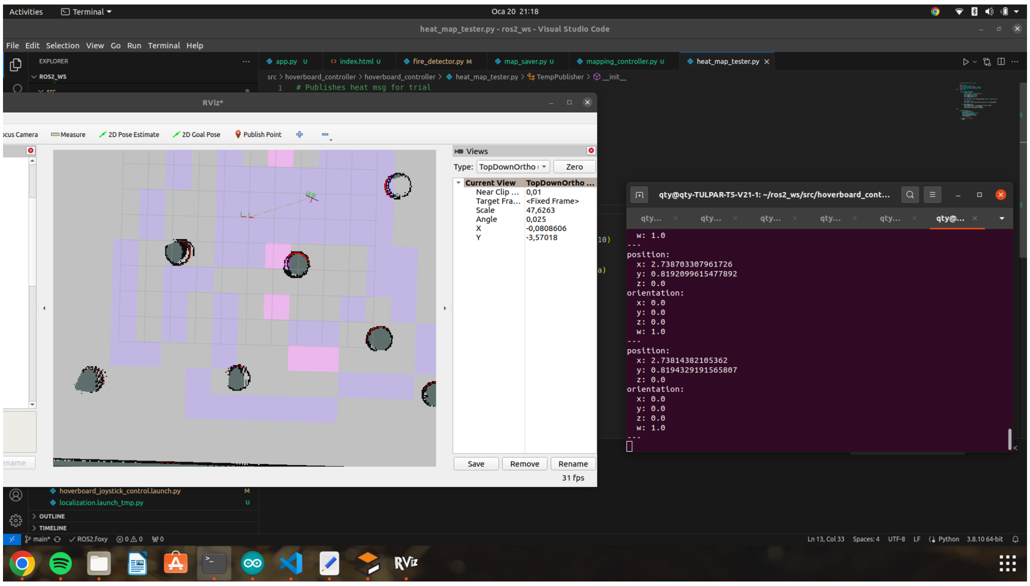

Figure 5.4 Heat Map Anomaly Detection

5.2 Real Life

5.2.1 Gmapping, Laser Scan Matcher and Heat Map Algorithms' Test

All the functionalities that are tested in Gazebo simulation environment is tested in real life with RPlidar as lidar and LM35 as the temperature sensor. All of the algortihms performed successfully.

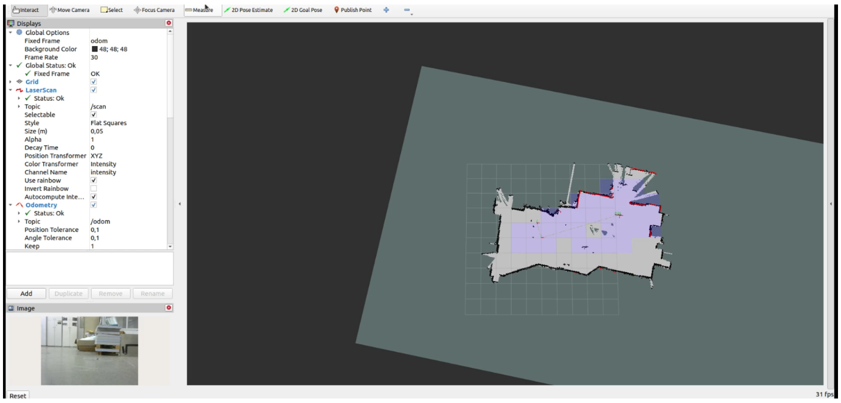

Figure 5.5 Mapping and Heat Map Algorithms' Test In Real Life

5.2.2 Fire Detection Test

The fire detection algorithm is tested with flash lights and a lighter. To make it produce false positive results the flash light is used. The fire detection algorithm successfully disregarded those input thanks to its fire sensor which only detects infrared light. The test is also tested with a lighter and the algorithm successfully detected the fire.

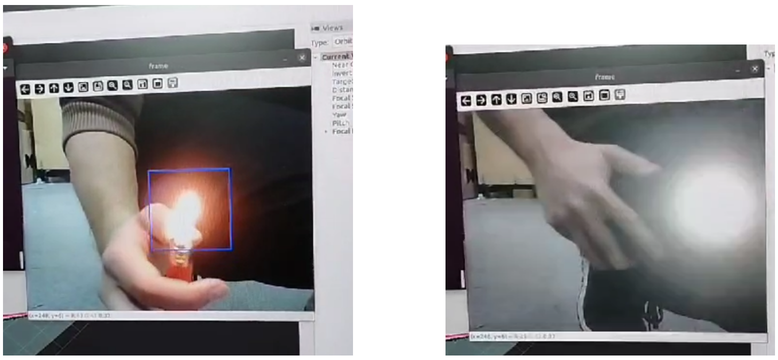

Figure 5.6 Fire Detection Results

6 Application

6.1 Web UI

A website is created as an user interface which allows users to reach live feed from an external camera, current temperature of the area which is gathered using a temperature sensor, robots real time location using lidar sensor and heat map of the area where robot is navigating from a web browser. The web application is built using Flask, a micro web framework for Python, and uses OpenCV for video streaming. From UI, user can toggle between video feed, map of the area and heat map.

When fire or a heat anomaly is detected an alert is raised and feedback is given to the user via UI. If the fire is detected with the fire detection algorithm from the camera, the result of the image processing is also shown to the user. The algorithm puts the detected fire in a blue square

There are two options that can be triggered from the UI: Mapping and navigation. The mapping option is chosen as default which starts the SLAM and Heat mapping algorithms. Mapping automatically chosen. A drop down menu is located at the top of the UI for choosing one of these options. When "Mapping" or "Navigation" is selected from drop down menu and "Start" button is clicked, web application triggers the related Python function. When "Mapping" option is selected, the robot will start to create a map and a heat map of the area. For this process, SLAM algorithm and Heat Mapping algorithm is used. If "Navigation" is selected, the robot will load the selected map files. AMCL algorithm is going to be called with the initial pose of x:0 and y:0. After loading the files and finding the robot position, all the functionalities can be used.

A map select menu is located Under the navigation option. In this drop down menu, user can access saved maps and load it in robots memory and website. After choosing a map, user can click on "Load" button to use the chosen map. User can also save a map by pressing "Save" button.

The computer that placed on the robot is served as a server. This application can be reached via computers IP and the port which is being used (for example 192.168.0.108:5000).

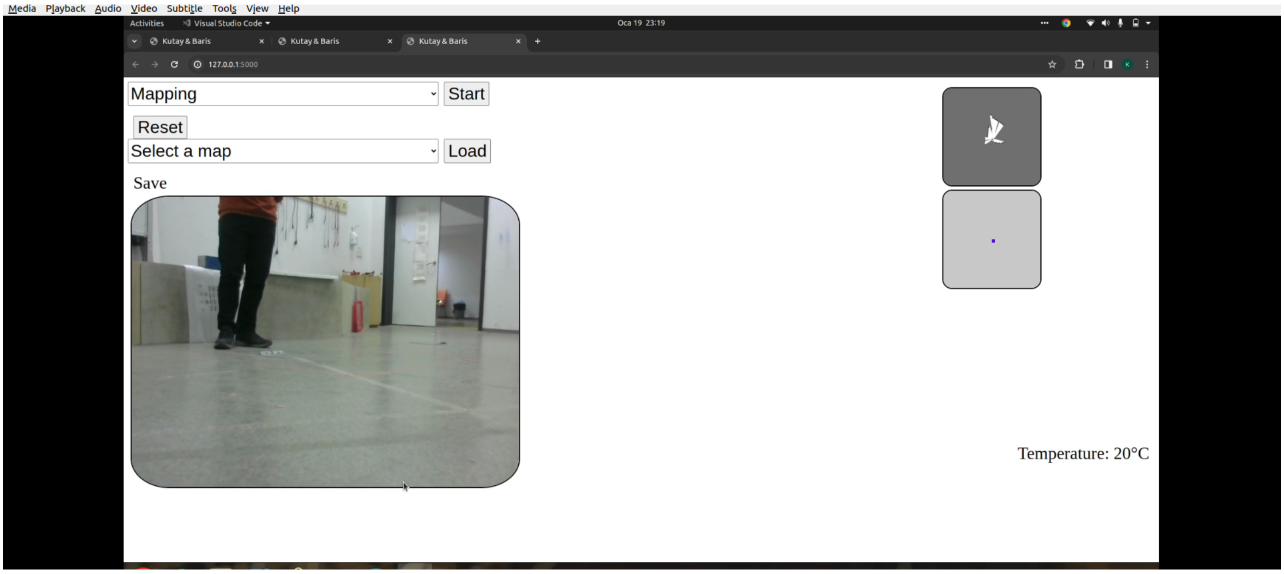

Figure 6.1 Website UI

7
Performance Analysis

7.1 CPU Performance Analysis

It is important to evaluate the CPU usage of the algorithms. Because there are couple of algorithms running simultaneously, it is important for them to be able to use enough CPU. Therefore the CPU performance is tested to see if a more powerful computer is needed. Slam gmapping and fire detection algorithms are observed to use the most CPU but the current hardware is able to support it.

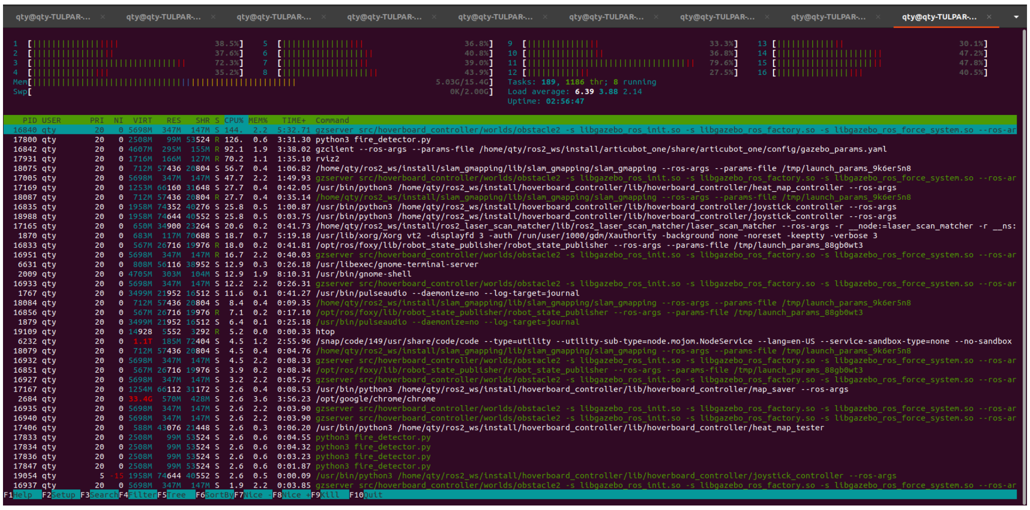

Figure 7.1 CPU Performance Analysis

7.2 Fire Detection Model Performance Analysis

It is crucial for the model to be able to work in dark and light backgrounds. It should be able to detect fire from the image. Though it is important not to have false positives, it is worse not to detect the fire. The false positives can be ignored by the use of fire sensor. However false negatives might cause the fire to be ignored and result in grave danger. Therefore, the threshold of the model is set to a lower value and the model is tested against 10 images that have light backgrounds and 10 images that have dark backgrounds. With the data collected the following graph is made. The model is able to detect fire on all conditions. However, it performs better in light backgrounds

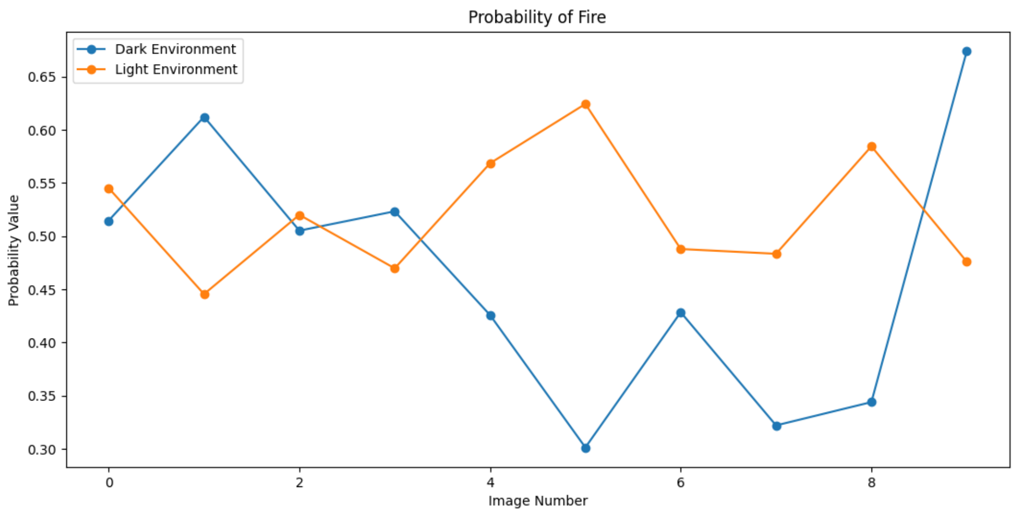

Figure 7.2 Fire Detection Model Performance Analysis

7.3 Robot Odometry Performance Analysis

It is important for the laser scan matching algorithm to perform as best as it can. Any drift in the odometry might cause the robot to be lost. Because all the maps (occupancy grid map and heat map) uses the odometry gotten from the laser scan matcher, the performance of the algorithm is invaluable. Therefore, the performance of the odometry data is tested. The robot is moved 1 meter and the odometry data is gotten from the laser scan matcher. Then those values are compared.

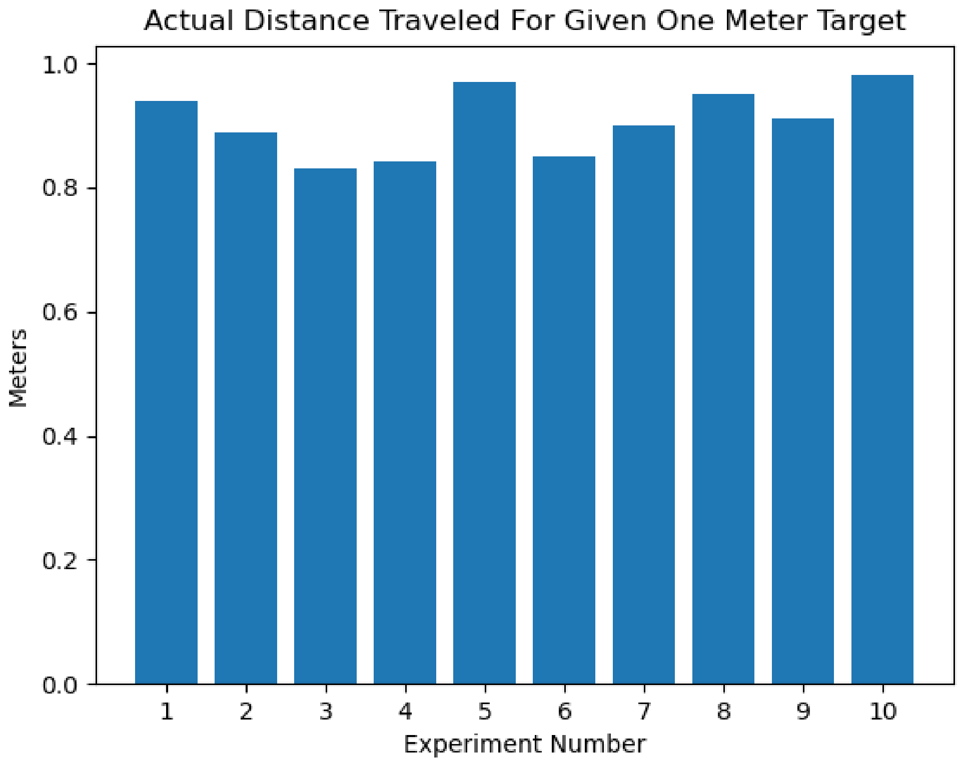

Figure 7.3 Odometry Performance Analysis

8 Conclusion

The project focuses on detecting fire with image processing, creating a occupancy grid map and heat map of the area and navigating inside the area with temperature, sonar and lidar sensors. SLAM algorithm is used, which enables robot to create a map of an unknown environment and help robot to find its locations in that environment that is mapped. The heat map is used to detect any temperature anomaly that might result in fire or damage. For detecting fire from camera, opencv library and infrared sensor is used. The task of fire detection and mapping is successfully handled by SLAM algorithm, Heat map algorithm and Fire detection algorithm.

The robot produced accurate maps in a closed area unless something shorter than the robot is on the same area. The problem with this is caused by the height of the lidar that is placed on the top of robot. To avoid this problem, everything shorter than the robot is excluded from the mapping area. This contributes to enhancing the overall accuracy and precision of the maps.

References

[1] NFPA. "Fire in industrial or manufacturing properties." (2018), [Online]. Available: https://www.nfpa.org/education-andresearch/research/nfpa-research/fire-statistical-reports/fires-in-us-industrial-and-manufacturingfacilities?l=110.

[2] A. S. R. Bishoyi *et al.*, "A deep learning approach for fire object detection in autonomous vehicles," *Journal of Physics: Conference Series*, vol. 2466, no. 1, p. 012 031, 2023. DOI: 10.1088/1742-6596/2466/1/012031.

[3] H. S. Sucuoglu *et al.*, "Development of mobile robot with sensor fusion fire detection unit," *IFAC-PapersOnLine*, vol. 51, no. 30, pp. 430–435, 2018. DOI: 10.1016/j.ifacol.2018.11.324.

[4] S. Li *et al.*, "An indoor autonomous inspection and firefighting robot based on slam and flame image recognition," *Fire*, vol. 6, no. 3, p. 93, 2023. DOI: 10.3390/fire6030093.

[5] X. Le, "Fire detection robot using type-2 fuzzy logic sensor fusion," M.S. thesis, 1970. [Online]. Available: https://ruor.uottawa.ca/handle/10393/32471.

[6] I. Yahaya, "Autonomous safety mechanism for building: Fire fighter robot with localized fire extinguisher," *International Journal of Integrated Engineering*, vol. 12, no. 1, pp. 304–314, 2020. [Online]. Available: https://publisher.uthm.edu.my/ojs/index.php/ijie/article/view/5731.

[7] E. Feru, *Hoverboard-firmware-hack-foc*, GitHub, 2019. [Online]. Available: https://github.com/EFeru/hoverboard-firmwarehack-FOC.

[8] R. M. Cole, *Fire detection from images*, https://github.com/robmarkcole/fire-detection-from-images, 2023.

[9] M. Quigley, K. Conley, B. P. Gerkey, and A. Y. Ng, "Ros: An open-source robot operating system," 2009. [Online]. Available: https://www.researchgate.net/publication/233881999%5C_ROS%5C_an%5C_opensource%20%5C_Robot%5C_Operating%5C_System.

Curriculum Vitae

FIRST MEMBER

Name-Surname: Barış BAKIM
Birthdate and Place of Birth: 05.02.2000, İstanbul
E-mail: baris.bakim@std.yildiz.edu.tr
Phone: 0507 485 19 87
Practical Training: Sade Yazılım ve Danışmanlık Company Software Development Department
Platin Bilişim Company Cyber Security Department
Entrapeer Company Software Development Department

SECOND MEMBER

Name-Surname: Anıl Kutay UÇAN
Birthdate and Place of Birth: 30.10.2001, Ankara
E-mail: kutay.ucan@std.yildiz.edu.tr
Phone: 0507 834 73 29
Practical Training: Polonom Robotik Company Robotics Department
Cezeri Company Robotics Department
Orion Company Software Development Department

Project System Informations

System and Software: Linux Operating System, Python, ROS, C
Required RAM: 4GB
Required Disk: 25GB