T.R.
ISTANBUL SABAHATTIN ZAIM UNIVERSITY
GRADUATE EDUCATION INSTITUTE
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INTELLIGENT COORDINATE DETERMINATION SYSTEM FOR AUTONOMOUS VEHICLES USING COMPUTER VISION AND WEB MINING

Ph.D. DISSERTATION

Erdal ALİMOVSKİ

Istanbul
January-2024

T.R.
ISTANBUL SABAHATTIN ZAIM UNIVERSITY
GRADUATE EDUCATION INSTITUTE
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INTELLIGENT COORDINATE DETERMINATION SYSTEM FOR AUTONOMOUS VEHICLES USING COMPUTER VISION AND WEB MINING

Ph.D. DISSERTATION

Erdal ALİMOVSKİ

Supervisor
Prof. Dr. Ahmet Emin KUZUCUOĞLU
Co-Supervisor
Assoc. Prof. Dr Gökhan ERDEMİR

Istanbul
January-2024

THESIS JURY APPROVAL

This study has been approved in partial fulfillment of the requirements for Ph.D. Degree in Computer Science and Engineering.

Supervisor: Ahmet Emin KUZUCUOĞLU

Member of jury: Mustafa Nizamettin ERDURAN

Member of jury: Burhanettin CAN

Member of jury: Zahra ELMİ

Member of jury: Oktay DOĞANGÜN

Approval by

Prof. Dr. Erhan İÇENER

Director, Graduate Education Institute

DECLARATION OF SCIENTIFIC ETHICS AND ORIGINALITY

This is to certify that this Ph.D. dissertation titled “Intelligent Coordinate Determination System for Autonomous Vehicles Using Computer Vision and Web Mining” is my own work, and I have acted according to scientific ethics and academic rules while producing it. I have collected and used all information and data according to scientific ethics and guidelines on thesis writing at Istanbul Sabahattin Zaim University. In both the text and bibliography, I have fully referenced all direct and indirect quotations and sources I have used in this work.

Erdal ALİMOVSKİ

ABSTRACT

Intelligent Coordinate Determination System For Autonomous Vehicles Using Computer Vision And Web Mining

Erdal Alimovski

Ph.D. Thesis, Computer Science and Engineering

Supervisor: Prof. Dr. Ahmet Emin Kuzucuoğlu

Co-supervisor: Assoc. Prof. Gökhan Erdemir

January-2024, 82 pages.

The robust autonomous navigation system relies on mapping, locomotion, path planning, and localization factors. Localization, one of the most essential factors of navigation, is a crucial requirement for a mobile robot because it needs the capability to localize itself in the environment. Global Positioning Systems (GPS) are commonly used for outdoor mobile robot application localization tasks. However, various environmental circumstances, such as high-rise buildings and trees, affect GPS, which leads to reduced precision or complete signal blockage. This dissertation proposes a visual-based localization system for outdoor mobile robots in crowded urban environments. The proposed system comprises 3 steps. The first step of the dissertation is to detect the text in urban areas using the "Efficient And Accurate Scene Text Detector (EAST)" algorithm. Easy, Tesseract, and Keras Optical Character Recognition (OCR) algorithms were applied to the detected text to find optimal ones by performing a comparative analysis of character recognition methods. In addition, the applied text detection and recognition algorithms were enhanced by applying post-processing and word similarity algorithms. In the second step, once the text detection and recognition process is completed, we pass the recognized word (label) to the Places API in order to return the recognized word's respective coordinates within the specified radius. By doing so, Points Of Interest (POI) data was collected for a particular area. Following this, the algorithm was developed to search the obtained data for the corresponding label and return the most appropriate coordinate through the Haversine formula. In the final step, a web-based map was developed to represent both the generated coordinates in the previous step and the coordinates from the GPS module.

The proposed system was tested in three distinct urban areas by creating five scenarios under different lighting conditions, like morning and evening hours. The outdoor delivery robot utilized in this dissertation was used for each scenario. In the experiments, it has been shown that the proposed system provides a low error of around 4 meters for localization tasks. Compared to existing works, the proposed system consistently outperforms all other approaches using only a single sensor. This indicates the efficacy of the proposed system for localization tasks in environments where GPS signals are limited or completely blocked.

Keywords: visual localization, GPS, delivery robot, text detection, text recognition

ÖZET

Bilgisayar Görseli Ve Web Madenciliği Kullanarak Otonom Araçlar İçin Akıllı Koordinat Belirleme Sistemi

Erdal Alimovski

Doktora Tezi, Bilgisayar Bilimi ve Mühendisliği

Danışman: Prof. Dr. Ahmet Emin Kuzucuoğlu

Eş Danışman: Doç. Dr. Gökhan Erdemir

Ocak-2024, 82 sayfa.

Sağlam otonom navigasyon sistemi; haritalama, hareket, yol planlama ve lokalizasyon içeren faktörlere dayanır. Navigasyonun en önemli faktörlerinden biri olan lokalizasyon, mobil bir robot için çok önemli bir gerekliliktir. Çünkü ortamda kendisini konumlandırma yeteneğine sahip olması gerekir. Küresel Konumlandırma Sistemleri (GPS), dış mekan mobil robot uygulamalarında lokalizasyon görevleri için yaygın olarak kullanılır. Ancak GPS, yüksek binalar ve ağaçlar gibi çeşitli çevresel koşullardan etkilenir ve bu da hassasiyetin azalmasına veya sinyalin tamamen engellenmesine neden olur. Bu tez, kalabalık şehir ortamlarında dış mekan mobil robotları için görsel tabanlı konumlandırma sistemi önermektedir. Önerilen sistem üç adımdan oluşmaktadır. Tezin ilk adımı, "Etkili ve Doğru Sahne Metin Dedektörü (EAST)" algoritmasını kullanarak şehir alanlarındaki metni tespit etmektir. Karakter tanıma yöntemlerinin karşılaştırmalı analizi yapılarak en uygun olanları bulmak için tespit edilen metne Easy, Keras, ve Tesseract Optik Karakter Tanıma (OCR) algoritmaları uygulandı. Daha sonra uygulanan metin tespit ve tanıma algoritmaları, son işleme ve kelime benzerliği algoritması uygulanarak geliştirilmiştir. İkinci aşamada, metin algılama ve tanıma işlemi tamamlandıktan sonra tanınan kelimenin ilgili koordinatlarını belirtilen yarıçap içinde döndürmek için tanınan kelimeyi (etiketi) Places API'ye iletiyoruz. Bunu yaparken belirli bir yarıçap içindeki ticari veya ticari olmayan bütün yerlerin (POI dataları) verileri toplandı. Ardından elde edilen verilerde kameradan tespit edilen etiketi aramak ve Haversine formülü aracılığıyla son alınan GPS verisine en yakın koordinatı döndürmek için algoritma geliştirildi. Son olarak hem önceki adımda oluşturulan koordinatları hem de GPS modülünden alınan koordinatların yansıtıldığı web tabanlı bir harita geliştirildi.

Önerilen sistem, sabah ve akşam saatleri gibi farklı ışık koşullarında 5 senaryo oluşturarak üç farklı kalabalık şehir ortamda test edilmiştir. Bu tezde her senaryo için dış mekan teslimat robotu kullanıldı. Yapılan deneylerde önerilen sistemimizden üretilen ve GPS modülünden elde edilen koordinatlar karşılaştırıldığında yaklaşımımızın diğer çalışmalara göre daha düşük hata oranı sağladığı gösterilmiştir. Bu, önerilen sistemin GPS sinyallerinin sınırlı olduğu veya tamamen engellendiği ortamlarda konum belirleme görevi için etkili olduğunu göstermektedir.

Anahtar Kelimeler: görsel konumlandırma, GPS, teslimat robotu, metin algılama, metin tespiti

ACKNOWLEDGEMENTS

I sincerely thank my supervisor, Prof. Dr. Ahmet Emin KUZUCUOĞLU, for his support, guidance, encouragement, and availability throughout the thesis period. In addition, I am thankful for how his expertise and perspective have greatly enriched my academic journey.

I extend my heartfelt appreciation to my co-supervisor, Assoc. Prof. Dr. Gökhan ERDEMİR, for his continuous support and guidance during this journey. I'm grateful for his availability, sincerity, and consistently friendly approach, whether working in the same laboratory or when he was kilometers away.

I extend my thanks to Prof.Dr. M. Nizamettin ERDURAN, director of IZU NARLAB, for his sincerity and guidance. He has provided technical equipment in the laboratory since I started working in NARLAB. Also, I appreciate his supportive, meaningful, and enjoyable conversations.

I would like to thank Asst. Prof. Dr. Oktay DOĞANGÜN, Asst. Prof. Dr. Zahra ELMİ, Asst. Prof. Dr. Haluk BAYRAM for their role as a jury member and for their support in the laboratory, which contributed to the thesis.

I would like to thank the esteemed Prof. Dr. İsmail KÜÇÜK for his continuous support. He always took the time to listen and offered solutions to my challenges.

I am grateful to my dear colleague and buddy, Asiye DEMİRTAŞ, for her constant encouragement, valuable contributions, and excellent collaboration during my Ph.D. Additionally, I appreciate her significant contributions to each part of the thesis and her always being by my side in both good and challenging times.

I would like to thank my dear friend Yağmur UŞAK, who always be there for any support, and her generosity in using her new car to transport the robot outside. I appreciate her creativity in taking us away from the laboratory to take a breath.

I would like to thank my dear friends Ömer Mutlu Türk KAYA and Seckin CANBAZ for their support in the mobile robot, hardware, and various IT assistance of the thesis and good conversations during my Ph.D. journey.

I extend my sincere thanks to Asst. Prof. Dr. Üzeyir Serdar SERDAROĞLU for his guidance that led me to postgraduate studies.

I extend my thanks to Asst. Prof. Dr. Ferhat ÖZOK and Asst. Prof. Dr. Aydın Tarık ZENGİN for the meaningful discussions and positive conversations during the thesis.

I sincerely thank all NARLAB members, including students and professors, for engaging in meaningful conversations during my Ph.D.

I express my sincere gratitude to advocate Hasan ŞAHİN for his support during my postgraduate studies.

I express my heartfelt gratitude to my cousin Hasan NECİP and his family for consistently being by my side and providing material and spiritual support.

I want to convey my heartfelt thanks to my beloved family, who consistently showers me with love, care, and unyielding support in all circumstances, with a special acknowledgment to my dear nieces Özge, Özgen, and Eymen Ali.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

API	: Application Programming Interface
CNN	: Convolutional Neural Network
CPU	: Central Processing Unit
CRNN	: Convolutional Recurrent Neural Network
CSLBP	: Center-symmetric Local Binary Patterns
CTC	: Connectionist Temporal Classification
EAST	: Efficient and Accurate Scene Text Detector
EKF	: Extended Kalmar Filter
Faster R-CNN	: Faster Regional Convolutional Neural Network
FFNN	: Feed-forward neural network
FPS	: Frame Per Second
GNSS	: Global Navigation Satellite Systems
GPS	: Global Positioning Systems
GPU	: Graphics Processing Unit
IMU	: Inertial Measurement Unit
JSON	: JavaScript Object Notation
LiDAR	: Laser Imaging Detection and Ranging
LOS	: Line-of-sight
LSTM	: Long Short Term Memory
MLPNN	: Multi-Layer Perception Neural Network
MP	: Mission Planner
NLOS	: Non-line-of-sight
NS	: Nearby Search
OCR	: Optical Character Recognition
POI	: Points of Interest
RAM	: Random Access Memory
RF	: Radio Frequency
RISS	: Reduced Inertial Sensor System
RNN	: Recurrent Neural Network
SDK	: Software Development Kits

SLAM	: Simultaneous Localization and Mapping
SM	: Sequence Matcher
UAV	: Unmanned Aerial Vehicle
UGV	: Unmanned Ground Vehicles
USB	: Universal Serial Bus

LIST OF SYMBOLS

K_m	number of the same characters in sequence
$\|S1\|\|S2\|$	represents the corresponding length for two strings
ϕ_1 and $\phi 2$	represents the latitude values at the two specified coordinates
$\Delta\phi$	represents the change in latitude
$\Delta\lambda$	represents the change longitude
$\Delta\lambda$	represents the difference in longitude between two points
R	represents the earth's radius
c	represents center angle between two points

CHAPTER ONE
INTRODUCTION

In recent years, outdoor mobile robots have become notably crucial due to their versatility and extensive range of applications. Outdoor mobile robots are employed for various tasks, including cargo delivery (Jiang, Song, & Zhou, 2022), planting and harvesting in agriculture (Yu, Zhang, & Liu, 2020), security and surveillance (Meddeb, Abdellaoui, & Houaidi, 2023), and supporting infrastructure maintenance (Lee et al., 2023). Navigation is the most important issue for mobile robots to accomplish their given tasks successfully (Ravankar, Ravankar, & Kobayashi, 2018). The success of the navigation systems relies on factors such as localization, mapping, path planning, and locomotion. Localization refers to the process of determining the robot's pose based on its environment, a previous point, or a given map. It can be done incrementally, where the position is tracked over time and changed in accordance with the robot's motions, or globally, where the pose is computed just once based on preliminary observations (Yasuda, Martins, & Cappabianco, 2020). Recently, numerous localization techniques were proposed for both indoor and outdoor environments. For outdoor environments, Global Positioning Systems (GPS), which is a satellite-based navigation system, forms a crucial part of Global Navigation Satellite Systems (GNSS) stand as the extensively utilized method in literature across a range of outdoor applications for determining precise locations of mobile robots.

Despite its widespread implementation, localization through GPS ensures accuracy declines or total signal loss due to several environmental factors. Common reasons for accuracy declines or total signal loss according to official U.S. government information about GPS and related subjects (UsaGovernment, 2023) are 1) blockage of satellite signals by big environmental factors such as skyscrapers, buildings, bridges, and trees; 2) reflection of signals from walls or buildings; and 3) both indoor or underground use. Three forms of GPS signal reception exist Line-of-sight (LOS), multi-path, and non-line-of-sight (NLOS) (R. Sun, Wang, & Zhang, 2020). In the LOS type, there are no obstructions or signal disturbances in the direct signal line between the satellite and the GPS receiver. This is the ideal situation to get the most reliable and accurate GPS

signals. Conversely, in multi-path reception types, the GPS receiver gets the direct LOS signals and the reflected signals from surrounding objects like buildings, trees, and bridges. These reflected signals can negatively interfere with the accuracy of the location data received by the receiver. In the context of GPS signal reception types, the signal can be totally blocked or less reliable in an NLOS scenario, where the receiver can only get the reflected signals. NLOS scenarios frequently occur in cities where a GPS receiver is located surrounded by towering structures. Comprehending the various types of GPS signal reception types is crucial to understanding the difficulties in achieving accuracy and dependability, especially in urban environments where the direct line-of-sight connection between satellites and GPS receivers is blocked. The challenges that arise in NLOS scenarios have an enormous effect on robotics, affecting robots' capability to navigate, perform the given tasks, and move effectively in environments where direct GPS signal reception is blocked. To overcome these challenges, it is necessary to develop cutting-edge sensor fusion systems, such as adding more sensors like cameras, lidars, or inertial measurement units. For example, by utilizing camera sensors, environmental information, such as street labels, shop names, and other contextual information, can be sensed, which can later, with a few processes, obtain the robot's actual position. Thus, it can significantly help overcome challenges associated with blocked GPS signals in urban areas.

Sensing environmental information, in other words, detecting and recognizing scene texts like street names, shop names, restaurant names, etc., in urban areas holds immense significance for mobile robots' navigation, intelligent traffic systems, visual assistance, and so on (Ranjbarzadeh et al., 2022). This kind of data can provide crucial contextual cues for that environment or place if it is properly processed. However, real-time text detection and recognition in urban areas is challenging due to the natural environment factors such as lighting, obstructions, weather conditions, and shooting angles, as well as large variability in scene characteristics in terms of text size, color, and background type (Yousef, Hussain, & Mohammed, 2020). Text detection, a prerequisite of text recognition, is a critical phase in textual information extraction and understanding. There are two types of methods for text detection in the literature: traditional and deep learning-based methods. In traditional methods, features are man-

ually intended to acquire the characteristics of scene text (Epshtein, Ofek, & Wexler, 2010; Yao, Bai, & Liu, 2012), whereas features are obtained entirely from training data in deep learning-based techniques (Netzer et al., 2011; Fleet, Pajdla, & Schiele, 2014). Since the text is accurately detected in real-time scenarios, it provides a clearer and more refined input to the Optical Character Recognition (OCR) system, resulting in higher recognition accuracy. The ability of OCR technology to recognize documents with a constant background color, the most basic fonts, and nicely aligned text is impressive. However, the performance in scene text recognition, such as street names, shop names, restaurant names, etc., is limited due to the complex backgrounds, distinct and distorted fonts, uneven illumination, and color variations (Zuo, Sun, & Mao, 2019). Developing an enhanced text detection and recognition system will improve the accurate recognition of street names, shop names, and restaurant labels. Obtaining this data type and integrating it with other fields, such as location-based web mining, augmented reality, and geo-tagging, will help for various applications, including mobile robot localization.

1.1 Problem Statement

Mobile robots have been used in various industries in the last decade, including logistics, medicine, agriculture, and healthcare. Navigation is one of the most crucial requirements for mobile robots to complete their given tasks successfully. A successful navigation system relies on various factors, but the key element is localization, which empowers the robots to determine their positions in the environment. In outdoor environments, Global Positioning Systems (GPS) have been extensively utilized in a wide range of outdoor applications. However, the accuracy of GPS-based localization is adversely affected by various environmental factors such as buildings, towers, trees, and bridges. As illustrated in Figure 1.1, there are three forms of GPS signal reception: LOS, multi-path, and NLOS. LOS signals are direct signals without any obstructions, ensuring accuracy. However, multi-path or reflected signals can negatively interfere with accuracy, potentially leading to incorrect localization. In NLOS scenarios, signals are either very unreliable or completely blocked. The obstruction of GPS signals, often caused by large and rigid objects or reflections from buildings and walls, can significantly decrease accuracy or result in a complete loss of signals. Considering the

limitations posed by signal blockage, it becomes imperative to emphasize the importance of developing a new system that does not rely solely on GPS signals. In cases where GPS signals are blocked, robot localization becomes crucial for ensuring accurate navigation and positioning. Therefore, developing a system that can determine the robot's position without relying on GPS signals is crucial.

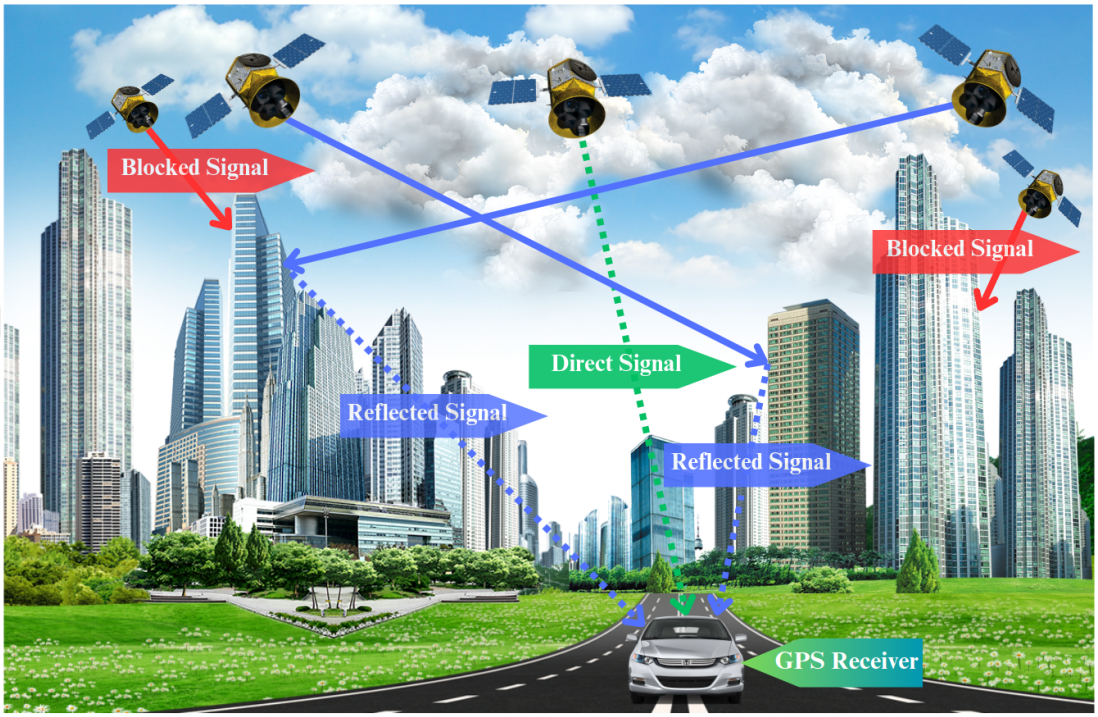


Figure 1.1: Problem illustration.

1.2 Related Literature

Localization is an essential requirement that allows the robot to determine its location, as well as features such as map creation and updating and optimal route planning(Yasuda et al., 2020). In general, the localization can be accomplished with the aid of GPS signals, which provide a global position, or by onboard sensors. However, GPS localization reliability is insufficient, especially in urban environments where buildings may obstruct or reflect satellite signals. To overcome these limitations, many different approaches have been proposed to solve the localization problem. Encompassing approaches based on vision, light detection, range (LiDAR), and inertial measurement unit (IMU), etc.

To address the localization problem (Georgiev & Allen, 2004) employed two methods. The first method includes a GPS module, odometry, compass, and tilt sensor. In ad-

dition, the Kalman filter was utilized to integrate the sensor data. The second method is based on a camera sensor, which estimates the pose. The pose estimation is accomplished by matching the taken building image with the smaller-scale facade models stored in the database. The results show that the first model's performance relies on GPS signals. They achieved remarkable results when the GPS module was connected to six or seven satellites. In contrast, the performance of the method decreased. On the other hand, experimental outcomes indicate that the second method performed well overall. However, its effectiveness becomes challenging when implemented in more extensive areas.

(Kümmerle, Ruhnke, & Steder, 2013) proposed a navigation system enabling the mobile robot to navigate in city centers. The system utilizes an extended SLAM routine process to complete its goal of handling the partially GPS-denied environments' outliers. For the localization task, the Monte Carlo localization approach is applied. Further, to deal with negative objects, terrain analysis methods were implemented. The experiments were carried out with an outdoor mobile robot called Obelix in the predefined area, spanning from the university campus to the Freiburg city center. The obtained results demonstrate the effectiveness of the proposed system. However, the accuracy of sensor measurements from the robot was compromised due to the presence of people in the surroundings, negatively impacting the localization task.

In (Vishal, Jawahar, & Chari, 2015), authors introduced a system that strengthens GPS localization by integrating vision algorithms and vice versa. The authors aim to improve the localization by utilizing an image retrieval algorithm. Inspired by the work (Roshan, Ardeshir, & Shah, 2014), in which an algorithm was developed to enhance user-specified GPS tags for images, the authors extended the system in this work. They applied it to discrete noisy GPS signals to get more accurate GPS signals. The experiments demonstrate the efficacy of the proposed system. However, it is essential to note that the system's effectiveness is contingent upon the reliability of GPS signals. The reliance on GPS signals might restrict the system's suitability to specific urban environments.

In (Qiao, Cappelle, & Ruichek, 2017), a visual localization method based on place recognition is introduced. This method realizes localization by recognizing recently visited places through the utilization of sequence-matching techniques. It operates by comparing query image sequences with a previously acquired image database. Moreover, a global GIST descriptor and a local binary feature CSLBP (Center-symmetric local binary patterns) are combined to create a multi-feature to improve the matching accuracy. Finally, Chi-square distance similarity measurement is employed for efficient sequence matching. The proposed system was evaluated on the Nordland dataset (Garg, 2020). The dataset includes video footage capturing a 728 km long train ride between two cities in northern Norway. The authors employed precision and recall metrics to evaluate the proposed method. The results show that the proposed system achieves a recall of more than 87% for spring-summer and spring-fall situations. The localization results also show that the suggested method can localize at least 60% of the test data. Experiments were conducted on simulation, paper does not include real-time localization experiments by the proposed system.

For accurate robot localization in hard outdoor environments, (Conceição, Neves dos Santos, & Costa, 2018) applied the Pozyx Algorithm, a hardware-based Ultra-Wideband system. In addition, EKF was performed. Thus, the output from the Pozyx algorithm was fed to EKF to fuse odometry with the Pozyx Range measurements. The experiments were carried out with AGROB v.16, a cost-effective outdoor robot. Since the obtained GPS data was not accurate and dependable, the authors used a Laser Scan as the ground truth to analyze the results. From the experiments, it can be concluded that the proposed EKF implementation presents better results than the Pozyx algorithm. However, it requires careful tuning to ensure proper convergence.

The authors (Sinha, Patrikar, & Dhekane, 2018) proposed a real-time CNN-based architecture, which integrates data from low-cost sensors on a mobile robot with information derived from images captured by a single monocular camera. The system utilizes an EKF to execute precise relocalization of the robot in real-time. The proposed approach begins with training a CNN to perform robot pose regression. This training is accomplished using RGB images captured by a monocular camera. The proposed approach

begins with the training of a CNN, which takes RGB images from the camera as input and applies regression for robot pose. Subsequently, the approach integrates the relocalization output obtained from the trained CNN into an EKF for robot localization. The system was tested in GPS-denied indoor and outdoor environments. The obtained results show that the proposed CNN-EKF approach is able to generate accurate localization. However, in environments characterized by repetitive scenes or visual content, the system faces difficulty providing accurate and simultaneous predictions for the robot's location.

In (Yousuf & Kadri, 2018), authors proposed two approaches for localization in indoor and outdoor environments. A Multi-Layer Perception Neural Network (MLPNN) was introduced to localize the mobile robot for indoor localization. For outdoor environments, the proposed system is conducted by sensor fusion. In the sensor fusion system, the collected data from the Inertial Navigation System and GPS module are fused using recursive state estimation and Kalman Filter. Then, the output from the Kalman Filter is combined with the odometer-based position data using a weighting scheme to estimate the robot's location. The results show that the proposed scheme can provide location in environments where GPS signals are available or not. However, the authors do not provide the system's accuracy or error rate. Additionally, the proposed system was evaluated in a simulation rather than a real environment.

To tackle the localization challenge, (Cai, Lin, & Kao, 2019) proposed a localization system that fuses the data obtained from GPS, IMU, and Visual Odometry. In addition, the system involves EKF. The proposed system was evaluated in two outdoor environments. The results obtained by the system were compared with the raw data from the GPS module to measure the system's robustness. Comparison results demonstrate that the proposed system exhibited an error rate of approximately 4 meters, in contrast to the GPS module, which showed an error rate of about 79 meters.

The authors proposed a low-cost-based approach for navigation mobile robots in both indoor and outdoor environments (Al Khatib, Jaradat, & Abdel-Hafez, 2020). The proposed system relies on fusing data from multiple sensors, especially employing low-

cost visual and inertial sensors. For the outdoor environments, an Extended Kalman Filter (EKF) along with GPS, wheel encoders, and a reduced inertial sensor system (RISS) was utilized to estimate the robot's position. Another EKF algorithm is introduced for indoor environments using a low-depth sensor called Microsoft Kinetic stream. Based on the experiments' findings, the proposed approach offers an acceptable performance to be deployed in real-time applications.

To localize the mobile robot in outdoor environments (Nilwong, Hossain, & Kaneko, 2019) proposed two methods based on deep learning and landmark detection. The first proposed method was based on Faster Regional Convolutional Neural Network (Faster R-CNN) landmark detection in the acquired image. A feed-forward neural network (FFNN) is trained to obtain the robot's location coordinates and compass orientation by utilizing detected landmarks. The second method performs a single convolutional neural network (CNN) to predict the location and compass orientation from the entire image. The experiments were conducted in two outdoor areas. The experimental results show that the Faster R-CNN average distance error was 28 meters, while the distance error of the CNN model was around 70 meters.

In (Wu et al., 2022), a visual localization approach was introduced, integrating depth and semantic information. The proposed approach employs semantic segmentation to capture a stable scene representation. It addresses variations in appearance between images caused by environmental changes by utilizing depth information obtained through depth prediction. The model was trained on the VKITTI 2 (Cabon, Murray, & Humenberger, 2020) and KITTI (Gaidon, Wang, & Cabon, 2016) datasets and subsequently evaluated on the Extended CMU Seasons and RobotCar Seasons datasets. The experimental results demonstrate that the proposed method exhibits remarkable performance in visual localization under various conditions, including weather, vegetation, regional environment, and illumination, based on evaluations conducted on the Extended CMU Seasons and RobotCar Seasons datasets.

In (Piasco, Sidibé, & Gouet-Brunet, 2021), a global image descriptor was introduced to address the challenges of image-based localization in demanding scenarios such as

cross-season, cross-weather, and day-night conditions. The proposed descriptor can handle visual changes between images by learning the scene's geometry. The strength of the proposed method lies in the fact that it only requires geometric information during the learning process. The proposed method was tested on the Oxford Robotcar public dataset (Maddern, Pascoe, & Linegar, 2017) and CMU Visual localization dataset (Bansal, Badino, & Huber, 2014). The proposed descriptor demonstrates remarkable performance in challenging cross-season localization scenarios, making it a valuable solution for long-term place recognition. Moreover, promising results are achieved in the context of night-to-day image retrieval.

In (Chen et al., 2023), an end-to-end DL-based visual localization algorithm was proposed. Initially, pre-processing tasks, including cropping, averaging, and timestamp alignment, are executed on datasets to minimize computational cost and time. Then, the processed dataset was fed to the proposed CNN-RNN-based model to identify the most impactful features for matching. Finally, the system predicts and provides output for the robot's current 3D translation and 4D angle information, thereby realizing a fully integrated end-to-end localization system. The proposed method was evaluated using three distinct datasets: the Cambridge Landmarks outdoor dataset (Kendall, Grimes, & Cipolla, 2015), the Microsoft 7-Scenes indoor dataset (Shotton et al., 2013), and the TUM Handheld SLAM dataset (Sturm, Engelhard, & Endres, 2012). During testing in the Cambridge Landmarks for outdoor environments, the proposed method exhibited promising results for localization. Nevertheless, there is currently a deficiency in implementing the proposed model for real-time applications in mobile robots.

In (Naseer, Burgard, & Stachniss, 2018), an innovative method for long-term robot localization using solely monocular image data is proposed. The proposed approach involves a unique data association technique to match incoming image streams with a stored image sequence in a database. Leveraging network flows, the method enhances localization performance by incorporating sequential information and maintaining multiple trajectory hypotheses simultaneously. Image comparison relies on a semi-dense description employing a histogram of oriented gradients' features and global descriptors from deep convolutional neural networks trained on ImageNet, ensuring robust

localization. The proposed approach demonstrates respectable performance in localization tasks through comprehensive evaluations across diverse datasets. The evaluation was carried out on datasets that were collected by driving through a city with a camera-equipped car during different seasons, including summer and winter.

1.3 Contribution of the Thesis

Previous studies focused on localization in urban environments have been valuable in shedding light on effective technologies and solutions. However, they have certain limitations. Most previous studies in outdoor localization, particularly in scenarios where GPS signals are denied, commonly employ multiple sensors or sensor fusion techniques. Additionally, despite implementing diverse techniques, the challenge of high localization error persists in urban environments for robots. The primary objective of this dissertation is to develop and implement a visual-based localization system for mobile robots in urban environments, particularly for areas where GPS signals are not accurate or totally denied. In addition, to represent the coordinates from the proposed system and GPS module, we developed a web-based map utilizing the Mapbox tool. Among the noteworthy contributions are:

- Comparison and analysis of EAST scene detection alongside various OCR algorithms in real-time on the university campus.
- Optimizing the EAST algorithm by utilizing post-processing and enhancing each OCR algorithm by applying a word similarity algorithm.
- Integrating the developed map with the proposed system to represent the generated coordinates and enable the map to operate offline.
- Obtain a comprehensive and up-to-date database that contains POI data.
- Verification of the proposed system's unproblematic and simultaneous operation in real-time cases where GPS signals cannot be received or are weak in urban areas with high-rise buildings.
- The performance analysis of the proposed system was tested in real-time utilizing the 8-wheeled delivery robot by creating five different scenarios in different

crowded urban environments such as crowded street, pedestrian street and commercial district .

- Analyzing the performance of the proposed system in different light conditions by performing experiments in both morning and evening hours.
- In contrast to prior research, which employed costly sensors or sensor fusion, in our study, localization was accomplished using only a basic webcam. This increases the usability of the proposed system.
- In contrast to previous studies that omitted the presentation of localization time for their systems, we showcase the localization timing of our proposed system.

1.4 Approach

This dissertation aims to develop an intelligent location-determination system for delivery robots based on computer vision and location-based web mining. Additionally, it aims to develop a map utilizing the MapBox tool to demonstrate the robot's location. The system's structure is as follows: performing text detection coupled with post-processing techniques; applying three different optical character recognition algorithms; enhancing the performance of OCR algorithms by utilizing word similarity algorithm; performance analysis of the OCR algorithms, localization-based web mining; creation of an algorithm to find most appropriate coordinate; development of a map; and finally performance evaluation of the determined location.

The EAST algorithm was performed to detect the texts in real-time urban environments. We apply non-maximum suppressions as a post-processing step to ensure more precise and efficient text localization and eliminate redundant or weak text detections. Once the text is detected, we feed it into Easy, Keras, and Tesseract OCR algorithms for recognition. In this part, we perform a performance analysis of the algorithms to select the more optimal algorithm for the proposed system by testing it on the university campus. In addition, to enhance the performance of OCR algorithms, we applied a word similarity calculation-based sequence matcher algorithm.

After the text detection and recognition process is completed in the previous stages, we pass the identified word (label) to the Google API to retrieve the recognized word's appropriate coordinates within the specified radius. In this way, the coordinates of places on Google Maps, such as all stores, shops, and markets within the specified radius, are saved in the database file with the .json extension. The obtained database will help the mobile robot localize itself when GPS and internet connection are unavailable.

When GPS and internet connections are unavailable, the developed algorithm will search the obtained database for the corresponding label and return the most appropriate coordinate by applying the Haversine formula. Besides this, the map was developed on MapBox to reflect both the generated coordinates in the previous step and the coordinates from GPS.

Finally, the proposed system was tested with the outdoor mobile robot in 3 different environments with various lighting conditions, such as morning and evening. The enhanced text recognition algorithm was used to localize the robot in real-time on urban streets. By applying localization-based web mining to the label data obtained from the enhanced algorithm, the robot determined its localization owing to the proposed system in cases where GPS signals are not received and there is no internet connection. The coordinates generated by the proposed system and those derived from GPS are displayed on MapBox. In the experiments carried out, it has been shown that the proposed system provides high precision and accuracy when the distance between both coordinates is calculated using the Haversine formula. Figure 1.2 illustrates the proposed system's flowchart.

1.5 Outline Of The Thesis

The remaining four chapters of the thesis are arranged as follows. Chapter 2 provides a comprehensive review of the methodologies and algorithms applied at each step in the detection and recognition of text on business signboards. Since text detection and recognition are the main focus of this chapter, scene detection, optical character recognition, and sequence matcher algorithms are discussed. Moreover, it offers comprehensive information regarding the experiments and outcomes of various algorithm fu-

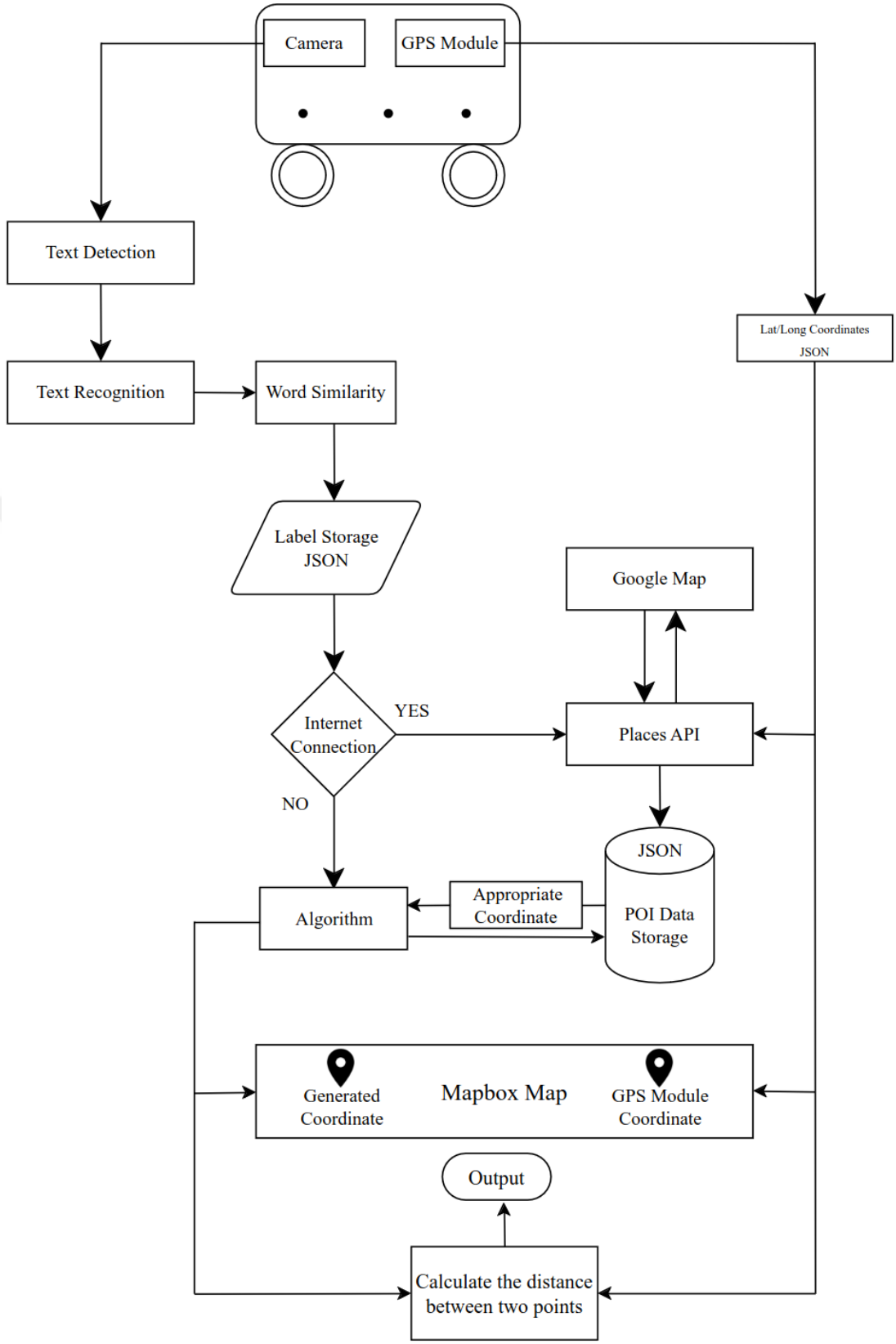


Figure 1.2: Flowchart of the proposed approach.

sions. Chapter 3 describes the tools such as Google API, MapBox, and the Haversine algorithm used in our proposed system when the GPS signal is weak or insufficient.

Furthermore, the system's working principle clarifies how we generate the coordinates in our proposed system. Chapter 4 demonstrates the experiments of the proposed system on a delivery robot, while Chapter 5 concludes the thesis, including possible future directions for research.

CHAPTER TWO
MATERIALS AND METHODS

This chapter will explain the mobile robot and sensors utilized in this dissertation. After investigating the existing literature, the EAST algorithm was applied for text detection. As a first step, the textual patterns in the live video were detected by EAST. After that, it was transmitted to Easy OCR, Keras OCR, and Tesseract OCR for text recognition. Furthermore, to enhance the OCR algorithm's efficacy, we employed the Sequence Matcher algorithm, which rectifies inaccurately identified labels. Finally, in this chapter, the performance of the OCR algorithms on a live video stream in urban areas was compared and evaluated by comprehensive experiments. Experiments were conducted on the university campus, where each signboard contained different textual patterns with fonts, backgrounds, colors, and board type. Based on the outcomes of these experiments, we selected the algorithms to be employed in the proposed system for visual position determination. In Figure 2.1, the flowchart of this chapter is demonstrated.

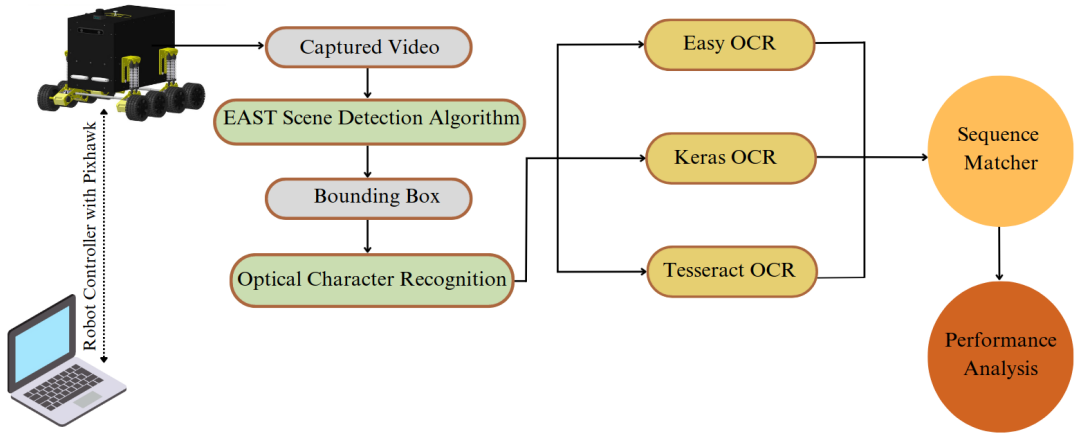


Figure 2.1: Flowchart of the proposed approach.

2.1 Data Collection

There is limited research in recognizing text in urban areas, such as streets, squares, and university campuses, using open-source OCR algorithms on live video streams. In this part of the thesis, live videos were captured on campus utilizing a Zed2i(Stereolabs, 2023) mounted at the top of an 8-wheeled mobile robot. In the conducted scenario, start and finish points were defined for the mobile robot. Thus, data were gathered in a specific area on campus where signboards, shops, and stationery are often used. In

addition, the fact that the signboard texts in that region were easily recognizable by the human eye played an essential role in choosing that region. Thus, in the testing phase, we could smoothly compare the recognized text by the algorithm with the actual text. Figure 2.2 presents the satellite image of the region, which contains the locations of each signboard on the test location. Also, the following path of the mobile robot is presented in the exact figure. Pixhawk controller(Pixhawk, 2023) was used for trajectory tracking and control of the robot. Existing text in the test area signboards are indicated in Table 2.1.

Figure 2.2: Area of data collection phase.

Table 2.1: Labels in signboards

Sigbnoard Nr.	Signboard Name	Existing Labels
1	Sabri Ülker Araştırma Merkezi	"Sabri", "Ülker", "Araştırma", "Merkezi"
2	KIRTASİYE STATIONARY	"KIRTASİYE", "STATIONARY"
3	TDV Book Store Kitabevi	"TDV","Book" "Store","Kitabevi"
4	Yesen Burger	"Yesen", "Burger"

As seen in Figure 2.3, each signboard contains different textual patterns with fonts, backgrounds, colors, and board types. For example, some shop names' characters have different colors and font types. All tests were performed under limited lighting conditions during cloudy weather. During experiments, the camera's resolution was

3840x1080 pixels, with 30 FPS, while the total video record duration was around 35 minutes.

Figure 2.3: Signboards samples in the test area.

2.2 Mobile Robot Platform

The 8-wheeled drive mobile robot designed in (Kaya & Erdemir, 2023) was used during this study's data collection and experiments. The mobile robot has eight rubber wheels with the ability to climb the pavements. Additionally, the robot has eight brushed DC motors. High gear-ratio-motor was preferred to prevent the robot from experiencing any problems due to ground type and unevenness in the outdoor environment. Quadrature encoders are seamlessly integrated into the rear of each motor to enhance precision and control.

The motors are controlled using an Arduino Mega 2560, a microcontroller board based on the Atmega2560. The motor driver board, which is capable of controlling all the motors on a single side, was utilized. This choice was made due to the Skid-Steering nature of the utilized robot. Consequently, we used two BTS7960 H-bridge motor drivers, each dedicated to controlling the motors on one side. After the determination

of the equipment of the robot, their power consumption was estimated. Consequently, the choice of the robot's battery was made based on the following criteria: battery type, power consumption of the robot, high capacity, battery voltage, and cost considerations. The lithium polymer battery was chosen for its high energy density, flexible design, low self-discharge rate, capability to handle high discharge currents, and long cycle life, making it well-suited for the application's specific requirements. The battery voltage level is higher than the nominal voltages of the motors. The battery voltage level exceeds the nominal voltages of the motors, with a nominal battery voltage of 14.8V compared to the motors' nominal voltage level of 12V. Additionally, Arduino boards, motor drivers, and the camera operate under different DC voltage levels. Therefore, the DC-DC buck converters were employed to reduce the voltage level.

Figure 2.4: The 3D views of the 8-wheeled mobile robot (Kaya & Erdemir, 2023).

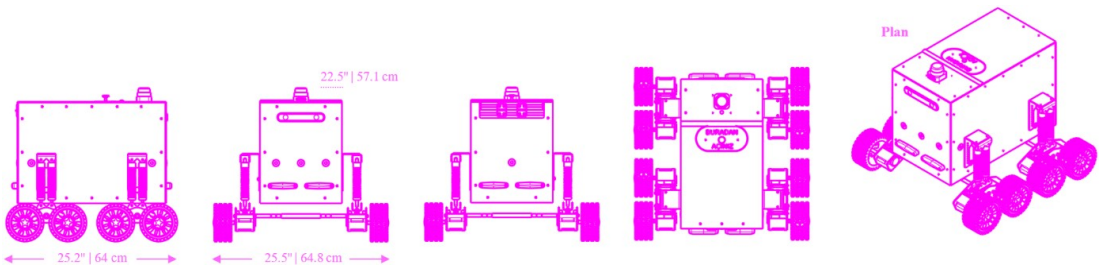


Figure 2.5: The views of the 8-wheeled mobile robot from different angles (Kaya & Erdemir, 2023).

The dimension of the robot is listed as total weight: 20 kg, width: 648 mm, length: 640 mm, height: 571 mm, and tire diameter: 140 mm (Kaya & Erdemir, 2023). Figure 2.5 demonstrates the 3D robot structure in different views. In the standard configuration of the robot, the camera was mounted on the top of the robot, as seen in Figure 2.6. When the camera's position was on the top of the robot, as in Figure 2.6, it would be a challenging task to successfully achieve text detection and recognition due to the high position of the signboards. Thus, the camera was set 1.08 meters above the ground after the modifications. The mobile robot after the modification is presented in Figure 2.6.

Figure 2.6: Mobile robot with the camera platform.

During the experiments in this chapter, the robot has a laptop configured with a 64-bit Ubuntu 20.04 Linux operating system. The laptop specifications include an Intel i5 processor, 8 GB RAM, and Nvidia's GeForce1680 graphic card, which we utilized along with GPU functionalities by installing CUDA and CuDNN libraries.

Turnigy 9XR PRO radio transmitter and V8FR-II module receiver control the robot remotely. The receiver module captures the transmitter signals and converts them into operational instructions for the right and left motors and motor drivers. Therefore, the robot can be controlled remotely with the Turnigy. It comes with eight channels. The Turnigy and his channels are depicted in Fig. 2.7.

Figure 2.7: The Channels of the Remote Controller

Each channel works as follows:

- **Channel 1:** Moving the stick up and down leads the robot's forward and backward movements.
- **Channel 2:** Moving the stick left and right leads the robot to turn right or left.
- **Channel 3:** The robot can only be accelerated using the left stick. The robot

moves more quickly if the stick is moved upward, while a downward movement results in slower movement and potential stops.

- **Channel 4:** None-used channel.
- **Channel 5:** It switches the robot's manual driving mode. In the low position, the robot operates in manual conventional driving mode. When the switch is in the high position, the robot switches to manual whirling mode.
- **Channel 6:** The potentiometer controls the spin and spinning direction of the robot. .
- **Channel 7:** Is utilized as an activation switch for each stick, switch, and potentiometer on the remote controller.

2.3 Sensors

The Zed2i camera was used to gather visual data from the urban environment for the experiments conducted in this chapter. In addition, the Pixhawk controller was utilized to control and track the robot's trajectory. However, for the experiments in the final system, the Logitech camera was used. The specifications of Zed2i and Pixhawk are given below. As well as the specification of the Logitech camera is provided as it was used in the experiments in Chapter 4.

2.3.1 Camera

Our thesis employed two distinct cameras: the Zed2i and Logitech C922. The Zed2i camera was utilized for the experiments carried out in Chapter 2, while the Logitech C922 was utilized on gimble in the final experiments. The specifications of both cameras are outlined below.

Stereolabs developed the Zed2i camera for image acquisition in various applications such as virtual reality, augmented reality, robotics, industrial automation, and many others. It is suitable not only for indoor but also for outdoor applications. The camera has a robust and reliable IR sensor that can provide high-quality images even in low-lighting conditions. The ZED2i captures both RGB and depth images in high resolution

(2K) with passive stereoscopic 3D technology based on a composite stereo image of the camera. The ZED2i camera is compatible with NVIDIA Jetson platforms and other ARM-based systems. This makes it appropriate for various applications, including mobile robots, UAVs, self-driving vehicles, and other intelligent devices.

The Logitech can capture videos at 1080p High Definition (HD) resolution with a 78-degree field of vision and maintain a frame rate of 30 frames per second. The Logitech C922 camera was selected for its many beneficial features, such as high resolution, auto-focus capability, lightness, and low cost.

2.3.2 Pixhawk Controller

Pixhawk controller provides significant convenience in autonomous driving in robotic systems such as unmanned aerial vehicles (UAV) and unmanned ground vehicles (UGV). Pixhawk is an open-source autonomous vehicle control system. It can provide a wide range of applications for robotic systems when integrated with ArduPilot software. Pixhawk comprises various modules, such as GPS, high-speed processors, and configurable interfaces. It can accomplish the navigation and trajectory control tasks crucial for mobile robots. To get the actual position of the mobile robot and perform trajectory tracking reliably, the U-blox NEO M8N GPS Module was mounted to the Pixhawk board. The GPS module starts searching for satellites and establishes a connection after a few minutes of being connected to the Pixhawk. Then, it sends the received location data packages to the board. Thus, the remote control station can wirelessly receive the location data thanks to 433 MHz Telemetry Radio modules. Mission Planner (MP) software was installed on the remote control station. Telemetry modules established the connection between the Pixhawk board and the remote PC. Hence, the robot was controlled and tracked by MP.

2.3.3 GPS Module

The Quectel L80 (Quectel, 2023) and Ublox Neo-M8 (Ublox, 2023) are both GPS modules designed to provide precise and reliable positioning information. The low cost of the Quectel L80 and Ublox Neo-M8 GPS modules makes them suitable for various applications. We measured the error of the Quectel and Ublox GPS modules for use in

this chapter and chapter 4. The measurements were carried out on the university campus at five distinct locations: in open areas, under the trees, and between the buildings. The error measurement was done as follows: Coordinates for the location where the GPS modules were placed were obtained from Google Maps. Following coordinates from the GPS modules were acquired. Finally, using the "measure distance" of Google Maps, the error of the modules was calculated. As seen in Table 2.2, for all locations, the error of the Ublox GPS module was less, except for location 2. Therefore, we decided to use the Ublox Neo-M8 GPS module.

U-blox Neo requires low power consumption and offers high efficiency in minimal acquisition time (Kamarudin & Tahar, 2016). The NEO-M8N GPS module provides the highest possible radio frequency (RF) performance and convenience of integration. Multiple GNSSs can be acquired and tracked by the device simultaneously. In ideal receiving conditions, the Neo-M8N GPS receiver ensures great position-tracking accuracy. The GPS module was modified to have a USB output directly connected to the computer during the experiments.

Table 2.2: Comparison of sensitivity of GPS modules

GPS Module	Location	Real Coordinate	Coordinate from GPS	Distance Error(m)
Ublox Neo-M8N	1	41.032792, 28.786023	41.032773, 28.786025	2.9
Quectel L80			41.032759, 28.786027	3.7
Ublox Neo-M8N	2	41.032666, 28.786378	41.032635, 28.786458	7.5
Quectel L80			41.032636, 28.786451	6.9
Ublox Neo-M8N	3	41.033545, 28.785261	41.033514, 28.785288	4.7
Quectel L80			41.033505, 28.785297	5.3
Ublox Neo-M8N	4	41.033645, 28.788009	41.033617, 28.788062	5.4
Quectel L80			41.033613, 28.788078	6.5
Ublox Neo-M8N	5	41.032539, 28.788987	41.032529, 28.788875	9.6
Quectel L80			41.032517, 28.788869	10.3

2.4 Scene Text Detection

Scene detection algorithms use image processing and artificial intelligence techniques to accurately and effectively detect text in complex scenes. These algorithms are used in many applications to detect and recognize, such as signage, logos, traffic signs, and security systems. Towards the literature review of scene detection algorithms, the EAST algorithm used in this study surpasses other algorithms in obtaining fast and accurate results. Therefore, it was decided to be used.

Scene text detection techniques have been accomplished on various benchmarks. These techniques, especially those that employ deep neural network models, have limitations when dealing with complex scenes. Interactions between the modules in the algorithmic model impact the text detection models' performance. Thus, text detection performance can be improved by utilizing a basic model that optimizes the loss function. Therefore, using a simple and effective EAST algorithm, text regions can be detected more rapidly and accurately (Zhou et al., 2017). The EAST algorithm was proposed by Zhou in 2017 to surpass the limitations in scene detection. The EAST algorithm employs a single neural network to predict the text's words or lines by avoiding candidate aggregation and word segmentation. The model architecture is demonstrated in Figure 2.8.

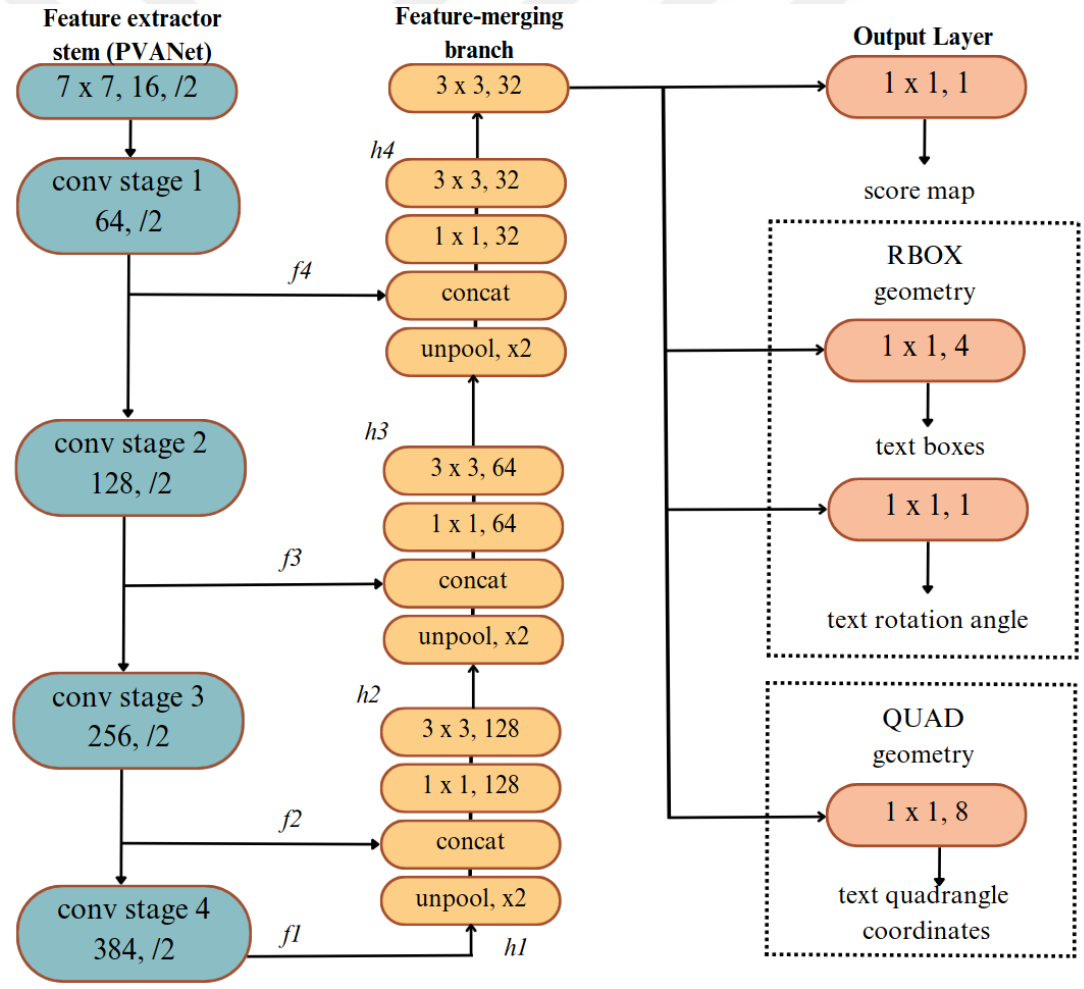


Figure 2.8: Architecture of EAST algorithm.

The model can eventually predict the 1-channel score map and the 4-channel box map if box data are tagged as RBOX. The algorithm can ultimately estimate the 1-channel score map and the 8-channel BOX map if the box data are annotated as QUAD. The EAST algorithm was trained using three distinct datasets: ICDAR (Karatzas et al., 2015), MSRA-TD500 (Yao et al., 2012), and COCO-Text. Due to the lack of a computer with sufficient computational resources, we employed a pre-trained EAST model for text detection.

2.5 Post Processing

Non-Max Suppression (NMS) is a post-processing method widely used in computer vision, such as object and edge detection. This method is used to organize rectangular boxes containing boundary text regions, which is also its intended use in our thesis. This post-processing technique also removes overlapping areas from the captured rectangular boxes inside the image. First, the proposal box with the highest confidence score is selected and added to another cluster (Khaki et al., 2020). Following that, the remaining boxes that have an excessive amount of overlap with the selected box are removed, and the box is removed from the P(Proposal) set. Other proposals are eliminated from the set P if their intersection over onion (IOU) is higher than a threshold that has been defined (Liao, Shi, & Bai, 2018). Consequently, this process generates an appropriate result set by sorting through multiple proposal boxes, acquiring the most reliable ones, and eliminating overlap.

2.6 Optical Character Recognition

As a first step, the EAST algorithm detected the text regions and taken into a bounding box. Then, the detected pattern was transmitted to three open-source OCR algorithms. These are Easy OCR, Keras OCR, and Tesseract OCR. In this study, the performances of these character recognition algorithms were compared. This process occurs again by selecting the most accurate and non-overlapping boxes and updating the set P until the set P is empty.

2.6.1 Easy OCR

Easy OCR is a cutting-edge OCR library that performs efficiently and supports over 80 languages. It was developed using Python and PyTorch frameworks. It executes detection using the CRAFT (Baek, Lee, & Han, 2019) method, a scene text detection model based on neural networks. As mentioned previously, the EAST algorithm was performed for scene detection in this study. Therefore, EAST and CRAFT algorithms were used for text detection before feeding the detected texts to the recognition phase. For recognition, Easy OCR uses a Convolutional Recurrent Neural Network (CRNN) (Shi, Bai, & Yao, 2016), which is comprised of three parts: ResNet for feature extraction (He, Zhang, & Ren, 2016), LSTM sequence labeling (Hochreiter & Schmidhuber, 1997), and Connectionist Temporal Classification (CTC) decoding (Graves, Fernández, & Gomez, 2006). Figure 2.9 depicts the exact Easy OCR working pipeline. This engine is among the best because of the pre-processing procedures included in this pipeline.

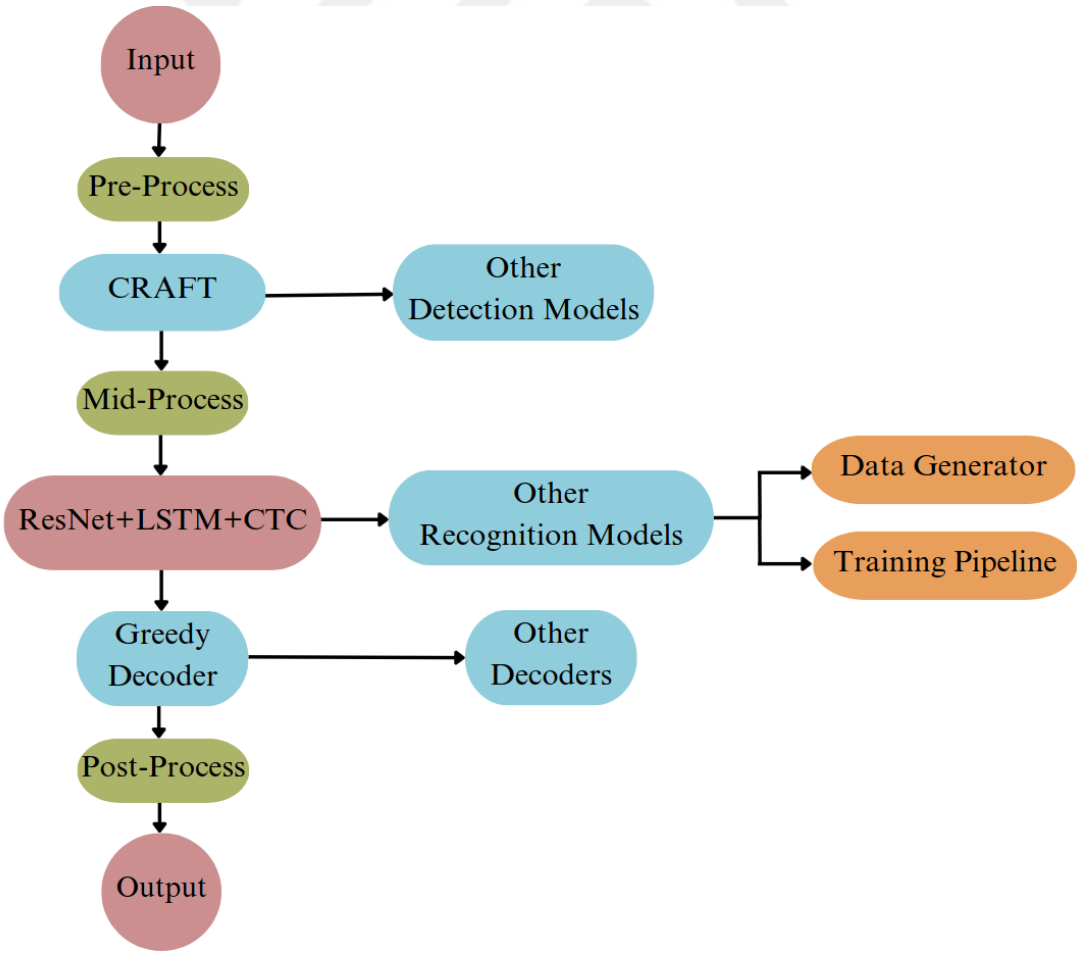


Figure 2.9: Architecture of Easy OCR algorithm.

This OCR engine can identify and detect more than 80 languages. We employed the algorithm in Turkish configuration since our dataset contains texts in Turkish. When the text is detected and taken into the bounding box, we feed it to the OCR engine to recognize the texts in the environment.

2.6.2 Keras OCR

Keras OCR is a deep learning-based OCR method in the Keras library (KerasOCR, 2023). Keras OCR offers a convenient interface for developing OCR models, which can identify text in various formats, including handwritten, printed, and even noisy or damaged text. Keras OCR has many typical applications, such as digitizing article texts, scanning and processing financial documents, and detecting texts in traffic signs, streets, and shops. Keras OCR is composed of two neural network architectures: CRAFT and CRNN. By investigating each character region and the affinities between characters, Keras OCR employs CRAFT to detect the text areas. Whereas for the text recognition phase, the original CRNN model was used. The architecture of Keras is presented in Figure 2.10. Due to the high accuracy of EAST text detection and some noisy detection of CRAFT during the implementation, we decided to disable the detection part of Keras OCR and utilize it just for character recognition.

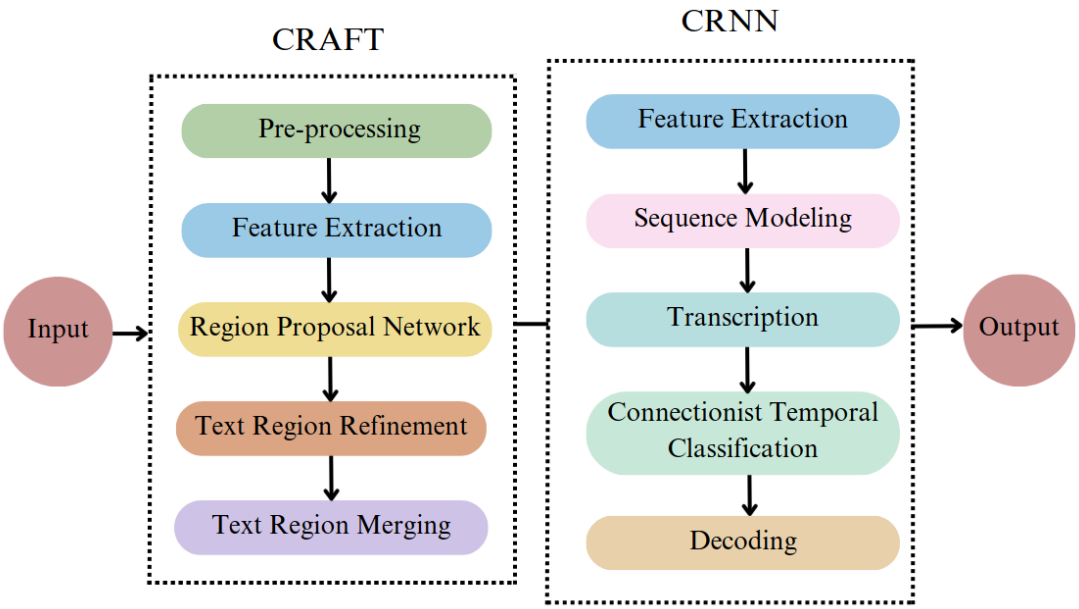


Figure 2.10: Architecture of Keras OCR algorithm.

2.6.3 Tesseract OCR

Tesseract is a well-known open-source OCR engine that Hewlett-Packard first developed and later supported by Google. Tesseract has been adapted for more than 140 different languages (Smith, 2007). Since the 4.0 version, a new engine built on Long Short-Term Memory (LSTM) was developed. Compared to prior versions of Tesseract, LSTM, a particular type of RNN, offers significantly improved accuracy. Additionally, the "pytesseract" Python library exists and offers quick access to this engine in Python. Figure 2.11 shows the Tesseract architecture and performed steps. The image was initially transformed into a binary image using an adaptive threshold. Following, character outlines were extracted by applying connected component analysis. Next, to organize the outlines into words, methods for character split and character association are performed. The two-pass word recognition process is ultimately carried out using clustering and classification approaches. To make its final determination regarding the recognized word, Tesseract consults both the language dictionary and the user-defined dictionary. Thus, the word with the smallest distance is given as an output.

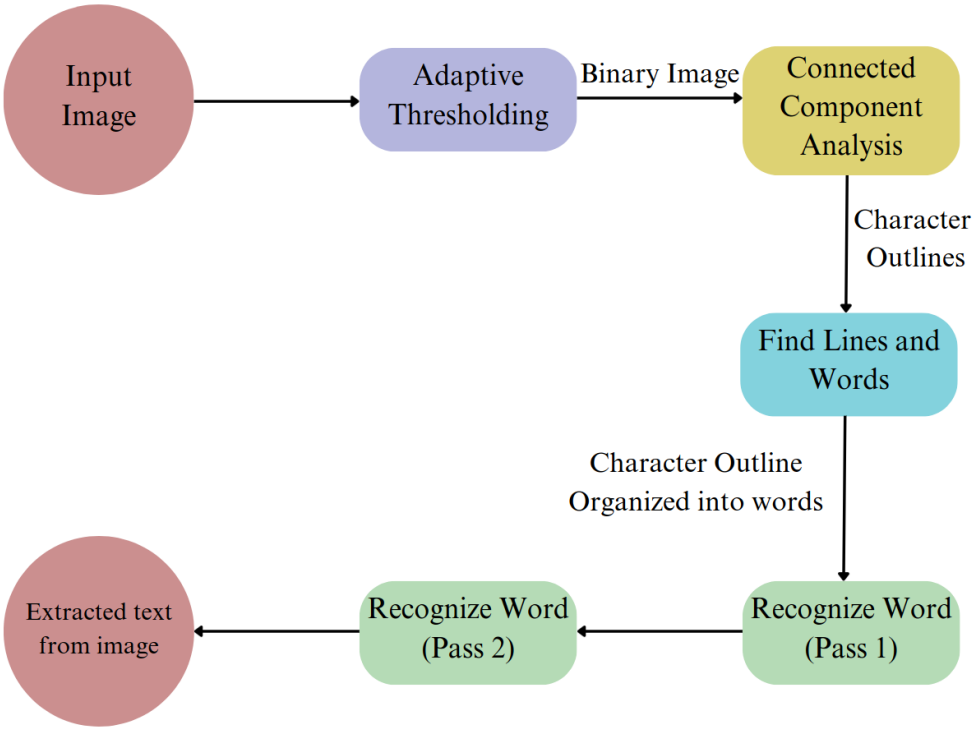


Figure 2.11: Architecture of Tesseract OCR algorithm.

2.7 Sequence Matcher

Sequence Matcher (SM)(Difflib, 2023) is a class of "difflib" module used to compare the similarity of two given strings. The Ratcliff/Obershelp algorithm (Ratcliff, Metzener, et al., 1988) is run in the background. After comparing the two given strings, the algorithm returns a score between 0 and 1. After comparing two strings, if the obtained score is greater than 0.7, it will be regarded as a keyword and stored as an actual word or label. The equation of the algorithm is as follows:

$$D_{ro} = \frac{2 * K_m}{|S1||S2|} \tag{2.1}$$

where K_m represents the number of the same characters in sequence, whereas $|S1|$ and $|S2|$ give the corresponding length for each of these two strings.

2.8 Comparison Results of OCR Models

In this study, once the video data was gathered with the mobile robot, the experiments were carried out on the computer with an Intel i5 Central Processing Unit (CPU), 16 GB random access memory (RAM), and a single Graphics Processing Unit (GPU) NVIDIA GeForce GTX 1680.

In the process of the experiment, firstly, the EAST algorithm was performed to detect the target scene texts. Thus, textual patterns in signboards, shop boards, and others have been taken into a bounding box. Secondly, the detected textual pattern was inputted into Easy OCR, Keras OCR, and Tesseract OCR algorithms for the character recognition step. Figure 2.12 shows a representative image of the test environment. To evaluate the performance of the OCR algorithms, we define the accuracy metric, the number of correctly recognized words divided by the total number of existing words multiplied by 100%. In addition, an analysis was conducted regarding how many times the existing words were recognized correctly by each algorithm and recognition time. Besides, the effect of the SM algorithm over OCR algorithms was investigated.

Figure 2.12: Illustration of the test area on the university campus.

In most cases, the EAST algorithm accurately detected the textual patterns in sign-boards, except in cases where the camera's distance was far away from the possible target texts. Its performance in cloudy weather and under limited lighting conditions is highly accurate, improving OCR algorithms' performance and correctness. In Figure 2.13, several accurately detected boards with text are presented. During the experiments, it was observed that when the angle of the mobile robot's camera was on the side position, the EAST algorithm was limited in detection. Therefore, as shown in Figure 2.14 the alignments of the bounding box vary.

Figure 2.13: A sample of textual pattern detection by the EAST algorithm in cloudy weather.

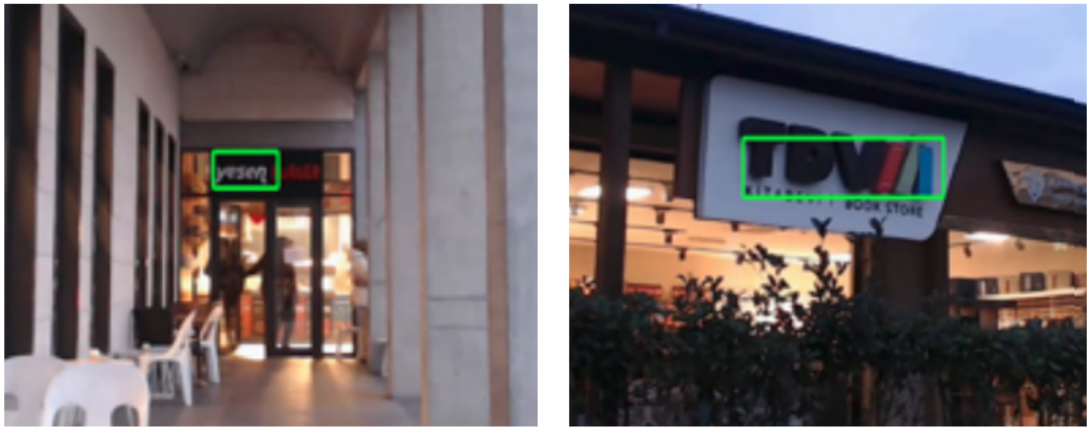


Figure 2.14: A sample of textual pattern detection by the EAST algorithm under limited lighting conditions.

When Table 2.3 is examined, it is clear that the Keras OCR algorithm outperformed other algorithms in accuracy. In addition, the average recognition time of the Keras OCR algorithm was the shortest compared to the different OCR algorithms. The algorithm recognized all the labels in the scene except the label "araştırma" even in bad conditions, such as low illumination and blurriness on the video stream caused by the vibrations of the robot. In contrast, the ability of the Easy OCR and Tesseract OCR to recognize the labels in urban areas was limited due to the variance of the label fonts. Thus, neither algorithm could recognize several labels such as "TDV" or "Burger".

Table 2.3: Performance analysis of OCR algorithms.

Methods	Average Recognition Time (second)	Accuracy(%)
Easy OCR	0.06	83.3
Keras OCR	0.04	91.6
Tesseract OCR	0.08	75

On the other hand, we measured the performance of the algorithms in terms of the number of repeats of each label and the recognition time of each label as listed in Table 2.4. Notice that recognition time means the time passed between text detection and recognition. Table 2.4 demonstrates that the Keras OCR algorithm recognized each label at least two or three times more than Easy OCR and Tesseract OCR. Besides this, recognition time is less than that of other algorithms for each label. Some labels, such as *Sabri*, *Kırtasiye*, and *Yesen*, were recognized many times, while labels such as *Stationary*, *TDV*, *Store*, and *Kitabevi* were less frequently identified. In some cases,

where the mobile robot was moved on a smooth ground surface, the vibration was less. Thus, the video quality was good, so the algorithms performed better due to the camera position alignment and type of surfaces. On the other hand, as the mobile robot was moved across the grass, the vibrations increased, and the video quality became blurry, making it harder for the algorithms to recognize the labels.

Table 2.4: Performance comparison of OCR algorithms

	Easy OCR		**Keras OCR**		**Tesseract OCR**	
Recognized Labels	Recognition Nr.	Average Time (second)	Recognition Nr.	Average Time (second)	Recognition Nr.	Average Time (second)
Sabri	10	0.05	16	0.04	3	0.08
Ülker	7	0.08	4	0.04	2	0.07
Araştırma	-	-	-	-	-	-
Merkezi	7	0.06	9	0.03	1	0.08
Kırtasiye	63	0.08	69	0.04	19	0.08
Stationary	1	0.07	4	0.03	1	0.07
TDV	-	-	4	0.03	-	-
Book	1	0.09	2	0.04	1	0.08
Store	1	0.06	2	0.05	1	0.09
Kitabevi	1	0.07	3	0.04	1	0.09
Yesen	16	0.10	160	0.05	75	0.08
Burger	2	0.08	71	0.05	-	-

During the experiments, it was observed that character recognition algorithms misidentified some labels by missing one or two characters of the actual labels. For example, algorithms recognized the actual labels which are *sabri*, *ülker*, *merkezi*, *Book*, and *Store* as ***abri***, ***ülke***, ***merkez***, ***boo***, and ***tore***, respectively. It is caused by the blurriness of the video stream and the camera's position. It was worth noting that, during the movement of the mobile robot, some vibrations occurred due to the ground type, affecting the quality of the video stream.

The SM method to address the shortcomings was applied in the previous step, which calculates the similarity between two words. The SM supported the output of each OCR algorithm, and it figured the word similarity between the recognized label and actual labels in a text file, which previously was stated. Thus, it suggested the most appropriate words. In our case, we set a threshold of 70%, so it only suggested labels higher than that percentage. Table 2.5 demonstrates the performance results of each algorithm after applying the word similarity calculation method. Compared with Table 2.4, it was observed that each algorithm's recognition performance increased for all the labels.

Table 2.5: Performance comparison of OCR algorithms with Sequence Matcher method.

	Easy OCR + Sequence Matcher	**Keras OCR + Sequence Matcher**	**Tesseract OCR + Sequence Matcher**
Recognized Labels	Recognition Nr.	Recognition Nr.	Recognition Nr.
Sabri	28	39	11
Ülker	19	16	7
Araştırma	6	7	1
Merkezi	20	27	4
Kırtasiye	89	99	35
Stationary	8	14	7
TDV	5	13	3
Book	7	9	2
Store	4	8	5
Kitabevi	8	10	4
Yesen	31	246	115
Burger	6	127	4

Table 2.5 demonstrates the performance results of each algorithm after applying the word similarity calculation method. The experiments show that word similarity methods positively affect character recognition algorithms, especially in limited labels. After examining the experiments, it can be concluded that Keras OCR combined with the EAST algorithm performs well in urban areas scenarios. However, Easy OCR and Tesseract OCR have some recognition and time limitations. After examining this experiment, we have decided to implement the EasyOCR algorithm in the final system due to its efficient accuracy and recognition time.

CHAPTER THREE
LOCALIZATION BASED WEB MINING

This chapter presents a detailed description of generating coordinates for localization tasks. In addition, the creation of the web-based map was explained. Furthermore, the developed algorithm for the overall system, encompassing various methods, is presented. Finally, the workflow of the proposed system is summarized before delving into the experiments.

3.1 Google Places Data

Google Places is a comprehensive service provided by Google which offers detailed information about places and locations all around the world. It includes an extensive database with geographic data, business listings, and points of interest (POI). POI data defines digital representations of specific locations or areas that interest some population segments (Sun, Hu, & Ma, 2023). POI data covers various types of locations, such as restaurants, markets, shops, grocery stores, schools, hospitals, and so on. POI data is pivotal for robotic localization, enabling them to navigate more efficiently for given tasks, particularly when GPS signals are limited or blocked. The Places API is crucial in providing necessary POI data for various locations. The Google Places extensive data can be accessed utilizing the Google Places API (NearbySearch, 2023), which essentially serves as a search function. There are four search techniques available for locations:

- **NS** provides a list of nearby places based on a given location; input location data type should be in terms of latitude and longitude coordinates.
- **Text Search** returns a compilation of locations in the region based on a search string, for example, ”Spaghetti.”
- **Place Details** returns more thorough details about a certain location, including user reviews.
- **Place Photo** enables the retrieval of photos associated with a specific place.

The Nearby Search (NS) is a crucial search technique for location-based services. It enables users to find POIs near a specified geographic location. Developers can customize search queries based on various parameters such as keywords, language, open now, rank by, and type. However, two fundamental parameters are required for the NS function: location and radius. Users should define a central "location" by providing latitude and longitude coordinates, which will serve as the epicenter of the search. The "radius" parameter defines the region where the NS will be conducted. It is a virtual boundary, limiting the search's range to a particular distance. Utilizing the "keywords" parameter is crucial in enhancing the search results. Searching by providing "keywords" or phrases as parameters is important for finding specific types of businesses, services, or any POI. Performing NS by using the parameters keyword, location, and radius is illustrated in Figure 3.1. The response from a NS provides extensive information on locations within a specified geographic area. This data usually includes each place's name, address, and geographical coordinates, allowing developers to present a dynamic and rich exploration of their environment.

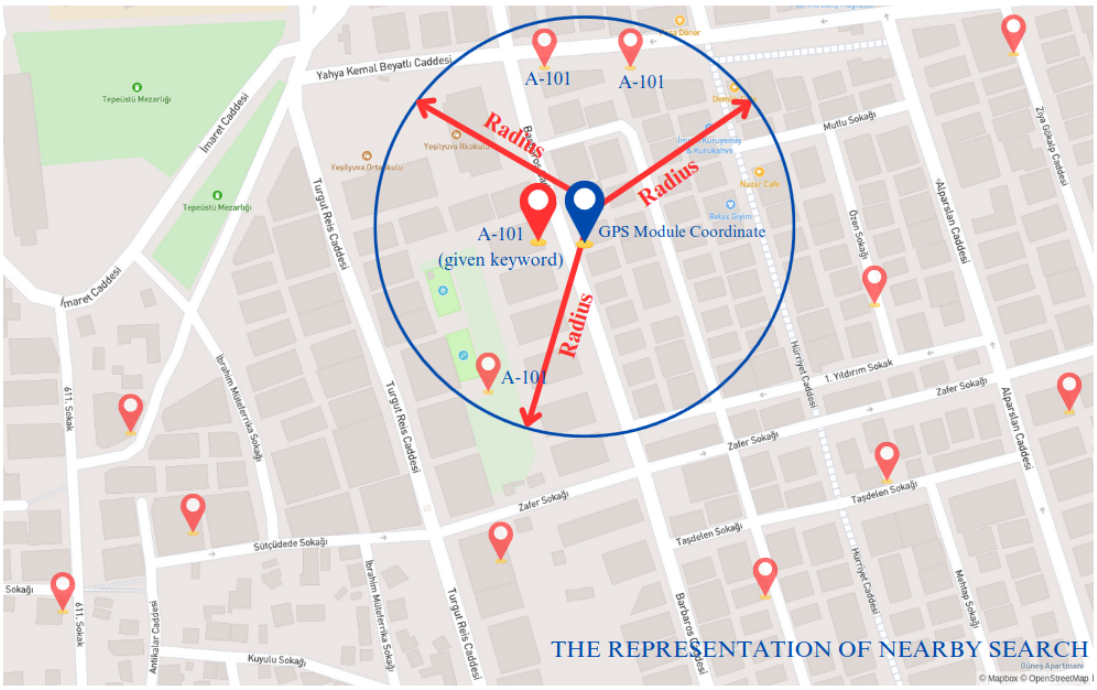
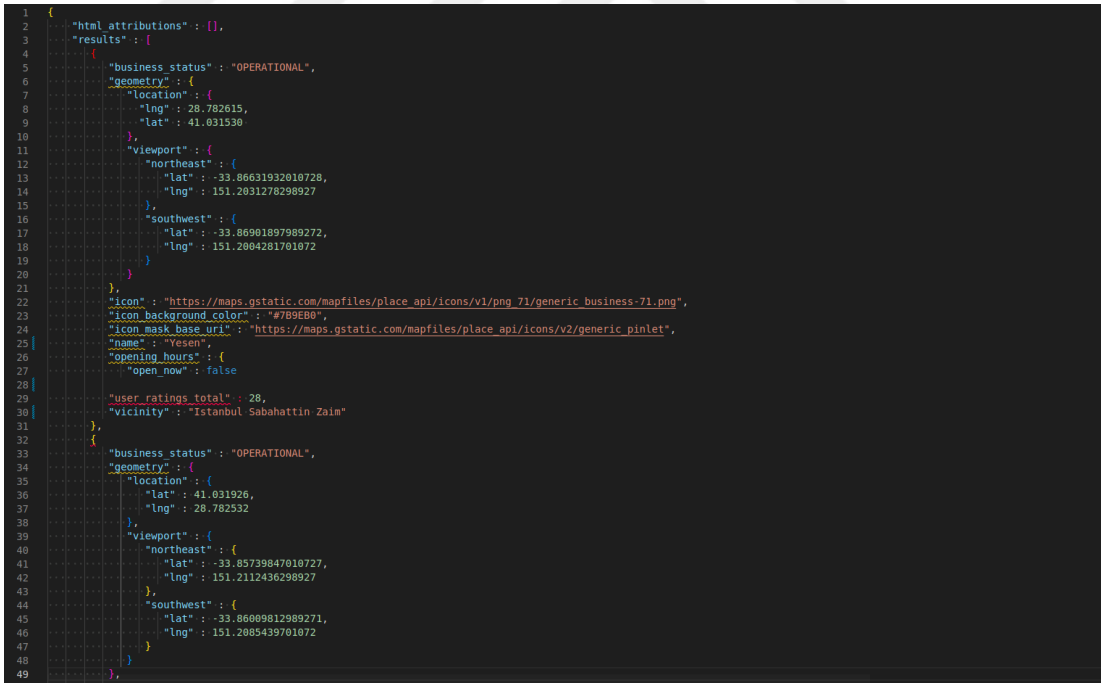


Figure 3.1: Obtained data

In this dissertation, we utilized the NS option with JavaScript programming language. In accordance with the scope of the study, it was intended to extract POI data from a certain region and save it to a file with a .json extension by utilizing NS. To employ the NS, firstly, we feed the coordinates from the GPS module placed in the robot as a

location parameter. As they represent the robot's actual position, the given coordinates will serve as the epicenter of the search. Secondly, we define a parameter "keywords" to specify the business names. This keyword parameter enables us to retrieve more precise information about the businesses in the region. After providing the location and keyword parameters, the last step is defining a radius to search for businesses in a specific region. Depending on the scenario, we set the radius between 300 to 500 meters. After applying all the steps, we collect POI data for particular regions where the experiments will be done regarding business name, coordinates in latitude and longitude form, business status, rating of the business, and so on. All the collected data is stored in a JSON file for further use in the final system. In the final system, where localization is the main goal, the crucial elements of each obtained POI data are the business names and their corresponding coordinates. The example of the returned data from the NS is demonstrated in Figure 3.2. As can be seen, the data includes location in the form of lat long, business name, business status, rating, address, type of place, and so on.



```
{
   "html_attributions" : [],
   "results" : [
      {
         "business_status" : "OPERATIONAL",
         "geometry" : {
            "location" : {
               "lng" : 28.782615,
               "lat" : 41.031530
            },
            "viewport" : {
               "northeast" : {
                  "lat" : -33.86631932010728,
                  "lng" : 151.2031278298927
               },
               "southwest" : {
                  "lat" : -33.86901897989272,
                  "lng" : 151.2004281701072
               }
            }
         },
         "icon" : "https://maps.gstatic.com/mapfiles/place_api/icons/v1/png_71/generic_business-71.png",
         "icon_background_color" : "#7B9EB0",
         "icon_mask_base_uri" : "https://maps.gstatic.com/mapfiles/place_api/icons/v2/generic_pinlet",
         "name" : "Yesen",
         "opening_hours" : {
            "open_now" : false
         },
         "user_ratings_total" : 28,
         "vicinity" : "Istanbul Sabahattin Zaim"
      },
      {
         "business_status" : "OPERATIONAL",
         "geometry" : {
            "location" : {
               "lat" : 41.031926,
               "lng" : 28.782532
            },
            "viewport" : {
               "northeast" : {
                  "lat" : -33.85739847010727,
                  "lng" : 151.2112436298927
               },
               "southwest" : {
                  "lat" : -33.86009812989271,
                  "lng" : 151.2085439701072
               }
            }
         },
```


Figure 3.2: An example of a POI data returned by NS API.

3.2 Mapbox

Mapbox (Mapbox, 2023a) is a web application that offers powerful tools and services for incorporating interactive and customizable maps. Mapbox provides various map-

ping services, including navigation, geocoding, spatial data analysis, etc. Mapbox creates compelling visual and dynamic maps by combining raster and vector tiles. Raster tiles provide detailed and high-resolution imagery for the customers. Raster tiles provide detailed and high-resolution imagery for the customers. In other words, they are pre-rendered images that present map information in a visually appealing format. A particular data storage scheme is defined for each tile set (Netek, Masopust, & Pavlicek, 2020). According to (Yang et al., 2011), schemes have a variety of parameters, including pixel size, tile form, and size, coordinate system origin, tile matrix size, and zoom levels. Raster tile maps were a novel concept at the start of the twenty-first century. However, they currently have a lot of drawbacks, mostly because they require generating a whole set of tiles each time the data geometry changes. Users cannot interact with the web maps on raster tile maps (Noskov, 2018). Google was the first to introduce vector tile maps, which are more recent than raster tiles. They were first used in the mobile version of Google Maps in 2010 and again in the web version in 2013 (Antoniou, Morley, & Haklay, 2009). Vector tiles are often saved on the server side and presented to the client as vector objects rather than pictures. Vector data do not contain any information for rendering; they are only points, lines, and polygons denoted by their vertex points. Mapbox offers a variety of map styles, and developers can choose from pre-designed styles or make their own using Mapbox Studio. Additionally, the platform provides software development kits (SDK) for other programming languages, opening it up to a large developer community. On the other hand, Mapbox offers a variety of map styles that developers can utilize to customize the appearance of their maps. Different map styles can be chosen Depending on the application's purpose, audience, and design preferences. Mapbox has offline map functionality. Thus, developers can create custom offline maps without requiring an internet connection. This is especially helpful when users might not always have internet connectivity, such as in remote areas or applications requiring reliable offline functionality. Depending on the requirements and usage of developers, Mapbox provides both free and paid services.

In this dissertation, a Mapbox-based web map was developed to demonstrate the coordinates obtained from the GPS module and those generated by the proposed system. The development of the map involves the incorporation of Mapbox GL JS, a powerful

JavaScript library designed for creating interactive and customizable maps. Utilizing MapBox GL JS enables us to render map data from Mapbox Vector Tiles, employing the Mapbox Style specification and hardware-accelerated graphics (WebGL). As a map style, we employed 'streets-v12'(Mapbox, 2023b), which is Mapbox's predefined style, offering a detailed representation of streets, landmarks, and geographical elements. In addition, Google Places was integrated with the map for utilizing Google Maps services. While developing the map, the map's initial view is centered at defined coordinates, with a zoom level set to 6. The map has been configured to operate offline without an internet connection. Once the map has been created, we define two points on the map. The first point, denoted in red, represents the coordinates obtained from the GPS module, while the blue point represents the coordinates generated from the proposed system. In the final system, as we will get coordinates sequentially from the GPS module and proposed system, the respective points will continuously update their location on the map. The example of the created map with the defined points is illustrated in Figure 3.3.

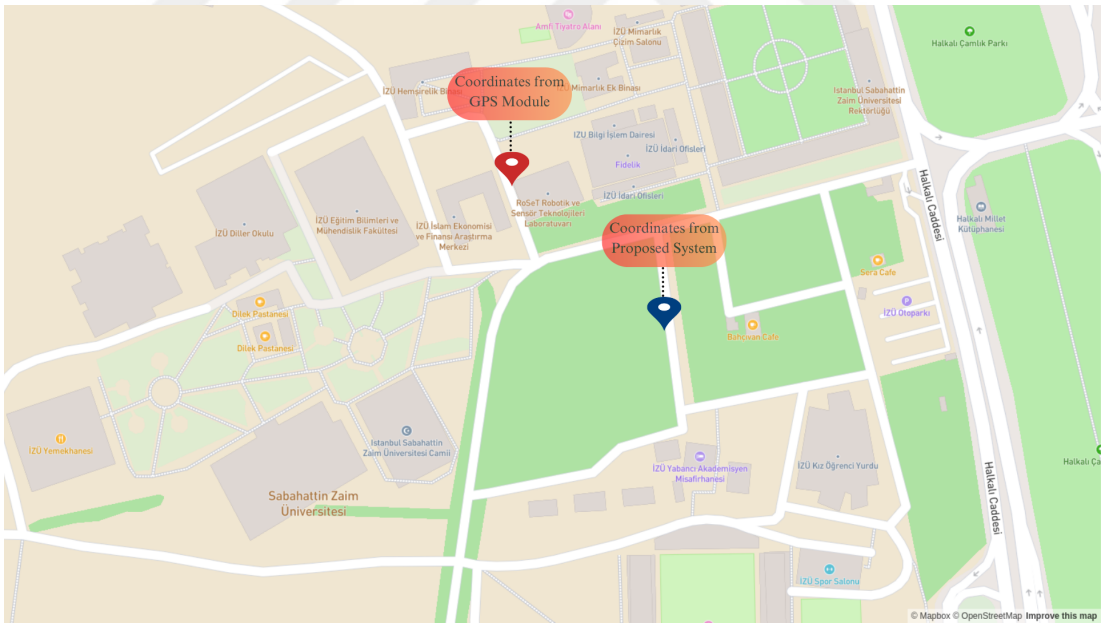


Figure 3.3: Illustration of the created map with example points on it.

3.3 Haversine

The Haversine Formula is a mathematical equation based on trigonometric principles used to calculate the distance between two points on the surface of a sphere, often applied to measure the great-circle distance between two locations on the Earth. The initial tabulation of Haversines was introduced by Andrew in 1805. However, Inman

officially coined the term “Haversine” in 1835 (Inman, 1849). The Haversine formula represents a specific instance of the law of Haversines, establishing connections between the sides and angles of spherical triangles. The Haversine distance (Sinnott, 1984), as illustrated in Figure 3.4, measures the great-circle distance in kilometers between two points on a sphere. The formula is defined as follows:

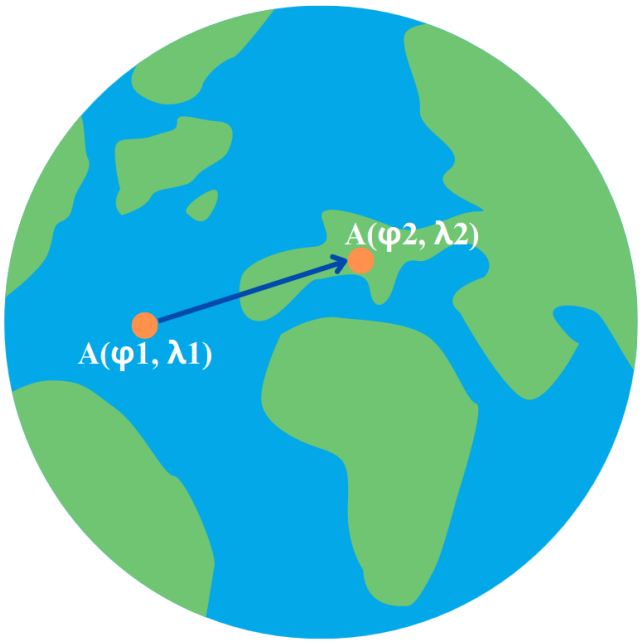


Figure 3.4: General representation of the Haversine formula.

$$a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right) \tag{3.1}$$

where ϕ_1 and ϕ_2 are the latitude values at the two specified coordinates, $\Delta\phi$ is change in latitude and $\Delta\lambda$ is change in longitude.

$$c = 2 \cdot \text{atan2}\left(\sqrt{a}, \sqrt{1-a}\right) \tag{3.2}$$

The center angle between two points is determined by equation 3.2.

$$d = R \cdot c \tag{3.3}$$

The shortest distance is determined by multiplying the value of R by c, where R corresponds to the Earth's radius and its approximately 6371 kilometers.

Considering previous research in the field, specifically regarding distance computation between points and subsequent path length determination, we will employ the Haversine distance instead of the Euclidean in this dissertation. We implemented the Haversine formula to calculate distances between GPS module coordinates and generated coordinates in this case. Furthermore, we integrated it into our developed algorithm to obtain the most relevant coordinate corresponding to a given label.

3.4 Developed Algorithm For Localization

The developed algorithm effectively manages the real-time tracking of both represented points on the map. Through systematic processes, the algorithm ensures the smooth functioning of the overall system. The red point on the map, in other words, coordinates obtained from the GPS module, are stored in a JSON file. The algorithm simultaneously retrieves these coordinates from the JSON file and illustrates the corresponding movements on the map in real-time. As the robot moves and recognizes the business's names to generate the respective coordinates of each business, demonstrate them on the map, and calculate the distance between two coordinates, the algorithm executes several processes.

All data obtained from NSes, particularly business names, are consistently stored in lowercase. However, the street business names vary, encompassing lowercase, uppercase, and a combination of both cases. The enhanced text detection and recognition system produces a string as its output. Depending on the shop names, the system output can be in lowercase or uppercase. As all the stored POI data is in lowercase, it is essential to convert the output from the recognition system to lowercase to facilitate an accurate search of the stored data and retrieve the coordinates to a respective business name. To accomplish this task, the algorithm employs the toLowerCase() function. Consequently, all outputs from text recognition will be converted to lowercase before searches are conducted on the JSON file.

As the robot moves and recognizes a business name, the algorithm searches the JSON file for respective POI data. For example, if the robot recognizes the business name “Migros,” the algorithm investigates the JSON file to find the ”Migros”-related data. If multiple “Migros” POI data exist, the algorithm uses the last GPS coordinates as a reference. Thus, by applying the Haversine formula, the most corresponding coordinate of “Migros” will be calculated and retrieved. The algorithm will retrieve the name of the business, in this case, “Migros” and its coordinates in latitude and longitude form. Doing so will demonstrate the coordinate on the map with the blue point. Consequently, both coordinates are visualized on the map. The Haversine formula is based on trigonometric functions, which normally require radians rather than degrees. Before applying the Haversine Formula to measure the distance between two points, we convert the latitude and longitude coordinates from degrees to radians with the following equation:

$$lat_{rad} = \frac{lat \cdot \pi}{180} \tag{3.4}$$

$$lon_{rad} = \frac{lon \cdot \pi}{180} \tag{3.5}$$

After converting the degrees to radians, we apply the Haversine Formula to calculate the distance between two points. The output distance is typically in kilometers, reflecting the Earth’s radius R used in the formula 3.3, which is approximately 6371 kilometers. Finally, to obtain the distance in meters, we convert from kilometers to meters as follows $meters = kilometers \cdot 1,000$.

3.5 System Overall

After implementing and developing the aforementioned methods and algorithms, we will summarize the proposed system in this dissertation in this section. Firstly, we carried out hardware modifications on the mobile robot utilized in accordance with the purpose of this thesis. Therefore, the robot became prepared for the experimental

cases. After examining the literature and conducting experiments for text detection and recognition in urban areas, we implemented the East algorithm for text detection and Easy OCR for text recognition. We apply the Sequence Matcher algorithm to correct wrongly recognized business names to overcome the challenges in real-time text recognition tasks.

In order to localize the mobile robot in cases where the GPS signal is low or totally blocked, we obtain POI data for particular areas by utilizing NS API. The POI data that was obtained was stored in a JSON file. Once the business name is recognized, the developed algorithm searches through the JSON file. It returns the corresponding POI data by applying the Haversine formula, specifically, the name of the shops and corresponding coordinates. In addition, to demonstrate and track the coordinates visually, a web-based map was developed using Mapbox. Subsequently, the coordinates generated by the proposed system and those obtained from the GPS module placed in the mobile robot are visually demonstrated on the map. Finally, to measure the sensitivity of the generated coordinate from the proposed system, we measured the distance between the coordinate from the GPS module and the generated coordinate from the proposed system. The pseudo-code of the proposed system is depicted in Algorithm 1.

Algorithm 1: Pseudo code of the Proposed System

while Robot Moving **do**
  $VideoDataset \leftarrow$ Record video in university campus
end while
Tmp=Load(VideoDataset)
$Algorithms \leftarrow$ {KerasOCR, EasyOCR, TesseractOCR}
for i=1 to $|Algorithms|$ **do**
  Test[i]=Algorithms[i](Tmp)
end for
for k=1 to $|Test[i]|$ **do**
  EasyOCR=OptimalAlgorithm(Test[i])
end for
$map \leftarrow$ (DevelopedMap)
$EAST \leftarrow$ Find text in the scene
$radius = 2km$
while Robot Moving **do**
  $video \leftarrow$ Video stream real-time
  $GPScoordinates \leftarrow$ Get coordinates from GPS module
  $BoundingBox \leftarrow$ EAST($video$)
  $post - processing \leftarrow$ BoundingBox
  $label \leftarrow$ EasyOCR.Recognize($post - processing$)
  $correctedLabel \leftarrow$ SequenceMatcher($label$)
  if internet connection YES **then**
    $ObtainPOIdata \leftarrow$ PlacesAPI($correctedLabel$, $GPScoordinates$, $radius$)
    $JSON \leftarrow$ ObtainPOIdata
  else
    $GeneratedCoordinate \leftarrow$ DevelopedAlgorithm($correctedLabel$).Haversine in JSON
    $map \leftarrow$ GeneratedCoordinate and GPScoordinates
    $DistanceBetweenTwoPoints \leftarrow$ Haversine(GeneratedCoordinate, GPScoordinates)
    $ProposedSystemsError \leftarrow$ convert kms to m (DistanceBetweenTwoPoints)
    if there is NO GPScoordinates **then**
      $DetermineRobotsLocation \leftarrow$ GeneratedCoordinate
    else
      $ProposedSystemsError \leftarrow$ GeneratedCoordinate and GPScoordinates
    end if
  end if
end while

CHAPTER FOUR
EXPERIMENTS

In this section, we will first introduce the robotic platform used in the experiments and the sensors on it. Secondly, the experiments conducted using the proposed system will be handled. Lastly, the proposed system will be compared with the existing works.

4.1 Experimental Platform

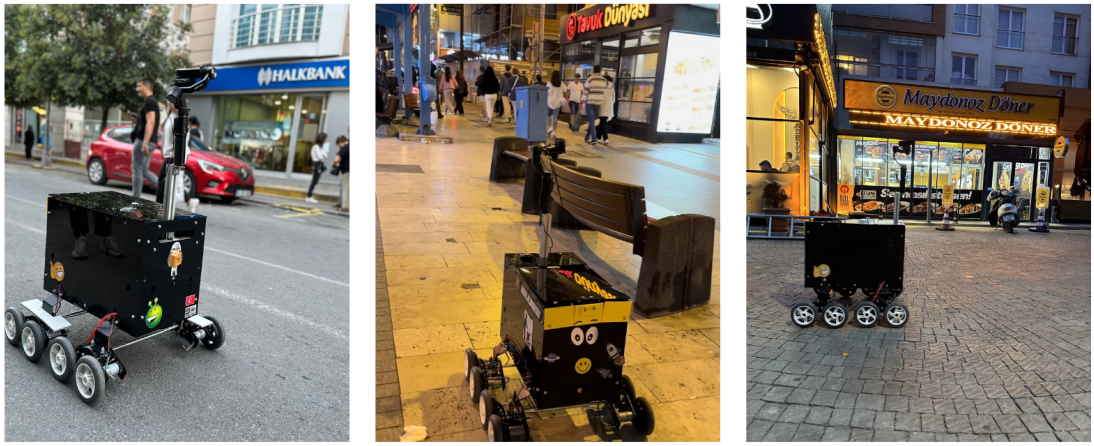
Before the experiments began, it was noticed that the Zed2i camera's physical size exceeded the capacity of the gimble head, causing undesired vibrations in the gimble. Therefore, it was decided to employ a Logitech C922 camera for the experiments.

For the default setup of the robot, the camera's position was out of angle concerning target views for detection. This situation could negatively impact the applied text detection and recognition algorithms. The platform was designed and mounted on the top of the mobile robot to bring the camera to the same angle as the target views. In the experiments carried out in Chapter 2, the video captured by the camera became blurred due to vibrations from the rubber wheels and the ground type. It has been noted that this has a negative impact on the algorithms' performance. To deal with this problem and reduce the vibrations, a gimble with a Logitech camera was placed on the top of the platform. Thus, the camera was set 1.08 m above the ground after the modifications. Additionally, throughout the experiments in this chapter, the robot was equipped with the same laptop utilized in Chapter 2. After the modifications, the final appearance of the robot is illustrated in Figure 4.1.

Figure 4.1: A view of the robotic platform from the laboratory environment

4.2 Localization Of Mobile Robot In Urban Environments

In this part, the experiments that were performed will be handled. To evaluate the accuracy of the proposed system and determine its capability to localize the robot in cases where GPS signals are lost, two types of scenarios were performed: 1) Obtaining coordinates from the GPS module and proposed system and 2) Obtaining coordinates from the proposed system. The coordinates from the GPS module are demonstrated on the map with a red point, while the coordinates from the proposed system are demonstrated with a blue point. The experiments were carried out on three distinct urban areas: Crowded Street (Area 1), Pedestrian Street (Area 2), and Commercial District Street 2 (Area 3). In Figure 4.2, the robot samples for each area are illustrated.



a) b) c)

Figure 4.2: Robot samples for each Area. a) Test Area 1, b) Test Area 2, c) Test Area 3

A crowded street operates as a one-way street for traffic movements, experiencing particularly heavy traffic, especially during the morning hours. The most common types of businesses along the street are banks and shops. The street surface includes only asphalt. The street with pedestrians operates as a thoroughfare exclusive to pedestrians and is closed to traffic. Commercial establishments, including shops and food stores, characterize the street. The street surface consists of tiles as well as asphalt in some regions. The crowded street 2 is open to two-way traffic and has pavement for pedestrians' use. The street includes a variety of businesses, such as coffee shops, markets, restaurants, and so on. The surface varies from region to region, including asphalt, concrete, and cobblestone. Considering the area in which the robot can move within the streets, the test area length of the experiments varies. The robot was generally operated at an approximate velocity of 5 km/h. However, owing to traffic jams and the presence of pedestrians in the experimental area, there were instances where it was necessary to either stop or reduce the velocity.

4.2.1 Localization in Crowded Urban Streets

The first experiment was conducted in a crowded urban street in test Area 1. As illustrated in Figure 4.3, start and end points were defined for the motion of the mobile robot. The length of the test Area 1 is around 130 meters. The street of test Area 1 is surrounded by 5 or 6-storey buildings. The surface of the street is asphalt. Thus, utilizing the gimble on the robot can avoid the vibrations caused by the surface, resulting in a good video stream. The numbers in Figure 4.3 represent the businesses that are

referred to in table 4.1. There are 8 businesses, 6 of which are banks, 1 shop, and 1 cargo shop. Each business consists of different signboards with its name with various types of fonts and backgrounds.

Figure 4.3: Start and End points of the robot for Test Area 1

Table 4.1: Businesses and their addresses in Test Area 1.

Number	Bussiness Name	Address
1	alBaraka	Cennet, Barbaros St. No: 73/B, 34290 Küçükçekmece/İstanbul
2	A101	Cennet, Barbaros St., No:73 34290 Küçükçekmece/İstanbul
3	YapıKredi	Cennet, Barbaros St. No:71, 34290 Küçükçekmece/İstanbul
4	Ptt	Cennet, Barbaros St. 69B, 34290 Küçükçekmece/İstanbul
5	HALKBANK	Cennet, Barbaros St. No:69, 34290 Küçükçekmece/İstanbul
6	VakıfBank	Cennet, Barbaros St. No:65B, 34290 Küçükçekmece/İstanbul
7	KuveytTürk	Cennet Barbaros St, No: 65/A, 34290 Küçükçekmece/İstanbul
8	Ziraat	Cennet, Barbaros Cd. No:63A, 34290 Küçükçekmece/İstanbul

Scenario 1

In this scenario, coordinates from the GPS module and generated coordinates are obtained to evaluate the performance of the proposed system. The experiment was conducted in the late afternoon; it began at 5.55 PM and ended at 6:01 PM. The robot's starting point is at the beginning of the business building marked with number one, and its ending point is at the building denoted by number eight in Figure 4.3.

As the mobile robot started moving, text detection and recognition were performed. In figure 4.4, from the a.) to the g.), the texts detected and recognized in real-time by the enhanced system are demonstrated, respectively, as mentioned in Table 4.1. When the video stream was handled, it was seen that the enhanced system could easily identify the label "alBaraka" from a variety of angles, even from the angles where sunlight exists, as can be seen in Figure 4.4 under option a.). As can be seen under option b.), the business, namely "A-101" does not have a signboard where its name is mentioned. The algorithm successfully recognized the business's name from the small label on the door. It was comparatively easier for the algorithm to identify the business name in options c.) and e.) due to the same background color and font used for businesses' signboards. In particular, "HALKBANK" was detected more often during the stream because of its name's capitalization. The enhanced system could not recognize the business name "Ptt" As can be seen from the option d.), the characters of the business name are adjacent to each other. So, the algorithm can not separate the characters, resulting in the interpretation of the name as a single character rather than individual characters. It has been noted that the algorithm performs more efficiently when handling business names (signboards) with dark color tones and white backgrounds. The algorithm frequently recognized the business "Vakıfbank" and "KuveyTürk" even when blurriness occurs in the video stream, as can be seen under options g.) and g.). The last shop, "Ziraat" was easily recognized from various angles due to its capital characters. In the overall process of detecting and recognizing text on the street in the 1st scenario, the enhanced system effectively recognized all business names along the route except one name. This success was achieved under various challenges such as lighting variations, image blurring due to the vibrations, and traffic jams.

a) b) c) d) e) g) h)

Figure 4.4: Detected and recognized business names by the proposed system in Scenario 1.

In Table 4.2, recognized business names (labels) with EasyOCR and corrected labels by Sequence Matcher with the similarity rate are presented. Here, the similarity rate means how many labels are similar to each other in the context of each character. In sce-

nario 1, Easy OCR recognized the business names “alBaraka,” “A-101”, “YapıKredi,” and “Vakıf” correctly. Thus, the similarity rate is 100%. The algorithm recognized the labels “HALKBANK”, “KUVEYTTÜRK” and “Ziraat” as missing one or two characters. Therefore, the similarity rate varies. The reason for missing one or two characters is that some look similar. For example, the algorithm recognized “HALKBANK” as “HAIKBANK”; here, the similar look of the characters ”I” and ”L” negatively affects the Easy OCR algorithm. Finally, as mentioned above, the algorithm could not recognize the label ”Ptt”. Therefore, it is demonstrated in the table with a dash (-). The obtained results show that utilizing the Sequence Matcher algorithm enhances the text detection and recognition system in Scenario 1.

Table 4.2: Recognized and Corrected Labels with the similarity rate in Scenario 1

Business Name	Recognized Label	Corrected Label with Sequence Matcher	Word Similarity Rate (%)
alBaraka	alBaraka	alBaraka	100
A-101	A-101	A-101	100
YapıKredi	YapıKredi	YapıKredi	100
PTT	-	-	-
HALKBANK	HAIKBANK	HALKBANK	87.5
VakıfBank	Vakıf	Vakıf	100
KUVEYTTÜRK	KuUEYT	KUVEYT	83.3
Ziraat Bankası	7iraat	Ziraat	83.3

Table 4.3 compares the GPS module coordinates and generated coordinates regarding distance error. In this context, distance error refers to the difference between the GPS module coordinate and the generated coordinate. In addition, the precision of the generated coordinate for each business was verified on Google Maps. We take GPS and generate coordinates for each business when the robot is in front of the business. The GPS and generated coordinates are taken when the robot is positioned before the respective shops. When table 4.3 is examined, it can be seen that the proposed system generated coordinates with 1.23 meters error when the robot was positioned in front of the business “YapıKredi” This signifies the lowest error rate. On the other hand, the highest error rate occurred when the robot was positioned near business “alBaraka” with a 10-meter error rate. When the robot approached the businesses ”A101”, “HALKBANK”, “VakıfBank”, “KUVEYTÜRK” and “Ziraat” the error rates were measured as 3.27, 3.24, 5.67, 7.62, and 00.00 meters, respectively. Consequently, the average error of the generated coordinates with respect to the coordinates of GPS module is 4.48

Table 4.3: Comparison of the GPS and generated coordinate in Scenario 1

Business Name (label)	GPS Coordinate	Generated Coordinate	Verification with Google Map	Distance Error (meter)
alBaraka	lat:40.991123 long:28.778287	lat:40.9910718 long:28.7782551	verified	6.05
A101	lat:40.991047 long:28.778276	lat:40.9910718 long:28.7782551	verified	3.27
YapıKredi	lat:40.990872 long:28.778350	lat:40.9908774 long:28.7783372	verified	1.23
PTT	lat:40.990768 long:28.778468	-	-	-
HALKBANK	lat:40.990647 long:28.778462	lat:40.9906612 long:28.7784283	verified	3.24
VakıfBank	lat:40.990503 long:28.778564	lat:40.9904952 long:28.7784983	verified	5.67
KUVEYTÜRK	lat:40.990434 long:28.778574	lat:40.9904682 long:28.7785097	verified	6.49
Ziraat Bankası	lat:40.99005 long:28.778730	lat:40.990093 long:28.778666	verified	5.43

meters. This shows the robustness of the proposed system, affirming its capability to uphold stability and performance in urban environments.

The proposed system generates coordinates once the business names are detected and correctly recognized. As illustrated in Figure 4.5, the generated coordinates and the coordinates obtained from the GPS module are demonstrated on the developed map. Note that the coordinates obtained from the GPS module are represented with red points, while generated coordinates from the proposed system are demonstrated with blue points. As the mobile robot moves, the red point on the map updates its location according to the coordinates from the GPS module, while the blue point is updated as the robot recognizes the business names and the system generates coordinates. When Figure 4.5 is examined from option a.) to h.), it can be seen that the blue point updates itself in synchronization with the red point as mobile robots come near to the businesses and recognize the business names. Except in option h.), where the label “Ptt” was not recognized, the system could not generate a coordinate and update the blue point. The effective updating of the blue point on the map along the 130-meter demonstrates the success of the suggested system. The timing between recognizing the business name, generating the coordinate, and demonstrating it on a map takes around 10 milliseconds on average. While the red point updates itself for 5 milliseconds.

Figure 4.5: Representation of GPS coordinate and generated coordinate on map in Scenario 1

Scenario 2

This scenario was carried out in the morning hours, under sunny weather conditions. The experiment starts at 09.06 AM and ends at 09.10 AM. Despite dozens of attempts

from various locations on the street, we could not get a signal from the GPS module due to environmental factors. Hence, in this scenario, while the mobile robot moves, it only updates its location on the map with the coordinates obtained from the proposed system. To measure the difference in the endpoint of the scenario, we manually define an end coordinate for the red point. Consequently, the robot will update its position based on the proposed system. At the end of the scenario, we will measure the distance between the manually defined coordinate and the robot's position. The robot's starting and end points are in contrast to scenario 1. In this case, the robot starting point is the business marked with the number 8, and the endpoint is the business denoted with the number 1 in Figure 4.3.

Due to the traffic jam, in contrast to scenario 1, we drove the mobile robot closer to the businesses in this scenario. As the mobile robots move, the detected and recognized business names are illustrated in Figure 4.6 sequentially based on their locations in the street. As depicted in Figure 4.6, from option a.) to h.), the enhanced system successfully recognized each business name except "Ptt". When the video stream was handled, it was seen that even if the weather is sunny and there are good lighting conditions, the camera angle negatively affects the system's recognition. Due to the business names being too high concerning the camera angle, the improved system encountered challenges in recognizing the name "VakıfBank"; thereby, it recognizes the related name in the last phase of the video stream, as can be seen in Figure 4.6 under option c). In addition, environmental factors such as moving vehicles introduced blurriness on the video stream in some cases, such as in option e). On the other hand, the rest of the business names were easily recognized dozens of times from different angles, as can be seen under options a), b), d), f), and g).

a) b) c) d) e) f) g) h)

Figure 4.6: Detected and recognized business names by the proposed system in Scenario 2.

In Table 4.4, it can be seen that Easy OCR, except for business names “A-101” and “alBaraka” encounters challenges in recognizing business names. The Easy OCR algorithm recognizes the “Ziraat” as “iraa”, “KUVEYT” as “KUUEYT”, and “Vakıf” as “akıfB”, “HALKBANK” as “IHALKBA” “YapıKredi” as “TapıKredi”. This is caused

by the camera angle and the traffic sequence, making it blurry. However, by utilizing Sequence Matcher, each wrongly recognized name was corrected. As a result, each business name was correctly recognized for the further process.

Table 4.4: Recognized and Corrected Labels with the similarity rate in Scenario 2

Business Name	Recognized Label	Corrected Label with Sequence Matcher	Word Similarity Rate (%)
Ziraat Bankası	iraa	Ziraat	80
KUVEYTTÜRK	KUUEYT	KUVEYT	83.3
VakıfBank	akıfB	Vakıf	80
HALKBANK	IHALKBA	HALKBANK	80
PTT	-	-	-
YapıKredi	Tapıkredi	YapıKredi	88.8
A-101	A-101	A-101	100
alBaraka	alBaraka	alBaraka	100

When table 4.5 is evaluated, it can be seen that as the mobile robot reaches the last business, the distance difference with the red point decreases to 7.15 meters. When the distance difference was evaluated for each business, it can be seen that the distance difference decreased simultaneously, which reveals that the proposed system creates smooth coordinates.

Table 4.5: Comparison of the GPS and generated coordinate in Scenario 2

Business Name (label)	GPS Coordinate	Generated Coordinate	Verification with Google Map	Distance Difference (meter)
Ziraat Bankası	lat:40.991024803056135 long:28.7783132543341	lat:40.9902392 long:28.7786062	verified	90.75
KUVEYTTÜRK		lat:40.9904682 long:28.7785097	verified	64.05
VakıfBank		lat:40.9904952 long:28.7784983	verified	60.90
HALKBANK		lat:40.9906612 long:28.7784283	verified	41.57
PTT		-	-	-
YapıKredi		lat:40.9908774 long:28.7783372	verified	16.51
A101		lat:40.9910718 long:28.7782551	verified	7.15
alBaraka		lat:40.9910718 long:28.7782551	verified	7.15

Upon correctly recognizing the business names, the proposed system generates coordinates. As it was mentioned before, we define constant coordinates for the red point due to the GPS signal blockage. Consequently, while the mobile robot is in motion and the proposed system operates in the background, the blue point continuously updates its position, as depicted in Figure 4.7. When the mobile robot reaches the end of the

route, the blue point comes closer to the red point. Figure 4.7 reveals that as the mobile robot reaches the end of the route, the blue point approaches the red point.

Figure 4.7: Representation of GPS coordinate and generated coordinate on map in Scenario 2.

4.2.2 Localization in Pedestrian Streets

The second experiment was carried out in the street with pedestrians, namely Test Area 2. Figure 4.8 presents the testing with the start and end points. The test area is approximately 105 meters long and obstructed by 6- or 7-story buildings. Most of the street consists of tiles, while a small part is asphalt. The numerical labels in Figure 4.8 correspond to businesses referred to Table 4.6. As seen in Figure 4.8, there are 7 businesses, and each business has unique signboards with various fonts and backgrounds.

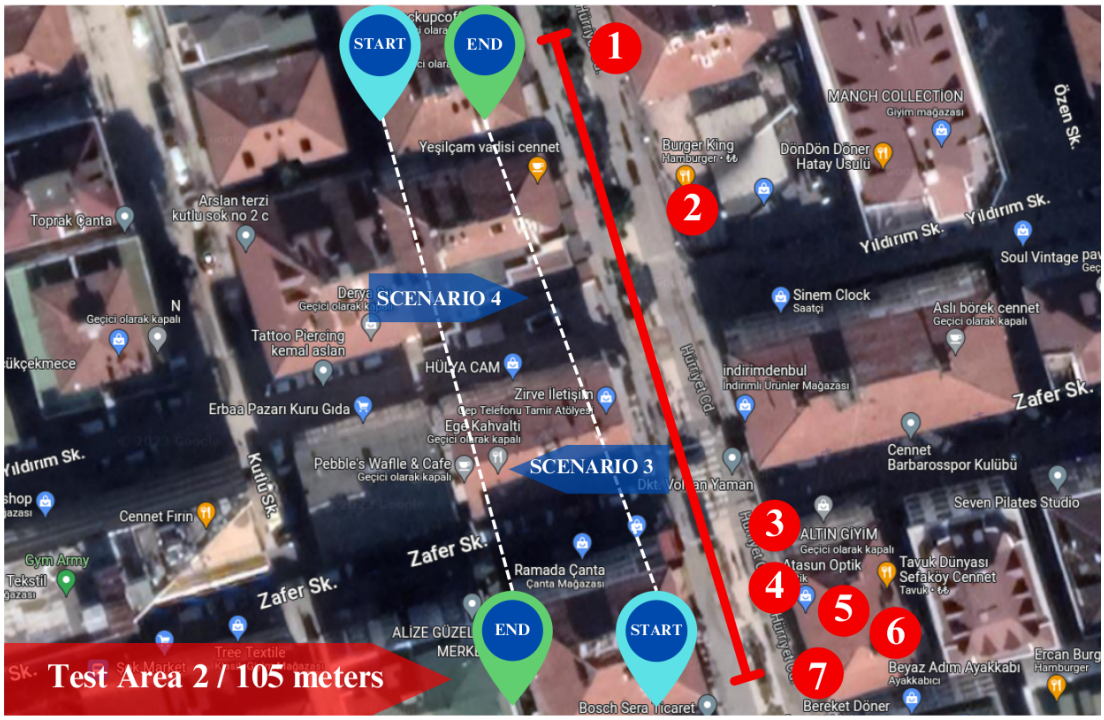


Figure 4.8: Start and End points of the robot for Test Area 2

Table 4.6: Businesses and corresponding addresses in Test Area 2.

Number	Bussiness Name	Address
1	Watsons	Cennet, Hürriyet St. No:62/B, 34290 Küçükçekmece/İstanbul
2	Burger King	Cennet, Hürriyet St. No:60, 34290 Küçükçekmece/İstanbul
3	Tavuk Dünyası	Cennet, Hürriyet St. No:56C, 34290 Küçükçekmece/İstanbul
4	Metin Kuyumculuk	Cennet, Hürriyet St. No:56, 34290 Küçükçekmece/İstanbul
5	GLORIA PERFUME	Cennet, Hürriyet St. No : 54 / A, 34290 Küçükçekmece/İstanbul
6	ERCAN BURGER	Cennet, Hürriyet St. No:52/C, 34290 Küçükçekmece/İstanbul
7	Atasun Optik	Cennet, Hürriyet St. No:52/B, 34290 Küçükçekmece/İstanbul

Scenario 3

The scenario took place in the evening, under the ideal weather conditions. The experiment began at 09.34 PM and concluded at 09.38 PM. The starting point of the robot is at the middle of the business building marked with number seven, and its ending point is at the building denoted by number one in Figure 4.8. In this scenario, we get coordinates from the GPS module and the proposed system.

Each detected and recognized business name is depicted in Figure 4.9, in sequence as their real location in the street. When the video stream was examined, it was observed that the enhanced system performs well even in the evening under limited lighting conditions. The enhanced system recognized the business names “Atasun Optik”, “ERCAN”, “GLORIA PERFUME”, “TAVUK,” and “KING” many times from different angles. Moreover, the system recognizes the business names “Metin” and “Watsons” a few times. Due to the handwriting name of the business and constant objects in the street, the system encountered challenges in detecting and recognizing the label “Metin”. Due to excessive light on the signboard, the system could not detect the label “Watsons”. However, it detected and recognized the business’s name from the advertising section above the signboard, as seen in Figure 4.9 under option g).

The recognized and corrected business names during scenario 3 are presented in Table 4.7. The Easy OCR algorithm misrecognized one character for the business names “ERCAN”, “Metin”, and “watsons” recognized them as “ERCAN”, “Justin” and ”atsons” respectively. Meanwhile, the rest of the business names are recognized without any difficulties. Due to the handwriting of the name “Metin” and the camera angle, the system failed to detect the character “M” properly, as can be seen in Figure 4.9 under option c). Thus, it was not fed into the OCR. In addition, due to the light in signboards, Easy OCR failed to capture the characters “N” in “ERCAN” and “w” in “Watsons”. Finally, each misrecognized name was corrected successfully with Sequence Matcher.

a) b) c) d) e) f) g)

Figure 4.9: Detected and recognized business names by the proposed system in Scenario 3.

For scenario 3, we consider the robot's actual position as generated coordinates due to the lack of GPS signal sensitivity. Therefore, we measure distance difference in terms of GPS module error. As shown in Table 4.8, the distance difference between the GPS coordinate and the actual position of the robot when the robot was near the businesses

Table 4.7: Recognized and Corrected Labels with the similarity rate in Scenario 3

Business Name	Recognized Label	Corrected Label with Sequence Matcher	Word Similarity Rate (%)
Atasun Optik	Atasun	Atasun	100
ERCAN BURGER	ERCA	ERCAN	88.8
GLORIA PERFUME	GLORIA PERFUME	GLORIA PERFUME	100
Metin	Jetin	Metin	80
Tavuk Dünyası	Tavuk	Tavuk	100
BURGER KING	KING	KING	100
watsons	atsons	watsons	88.6

"Atasun Optik", "ERCAN BURGER", "Metin" was around 43.25, 31.72, and 27.63 meters, respectively. Further, the GPS signal improved its sensitivity when the robot approached the businesses "GLORIA PERFUME" and "Tavuk Dünyası" resulting in a reduced distance difference of approximately 13 meters. In the last two businesses, we faced GPS blockage. Therefore, we don't have any measurements. This scenario shows the importance of developing a localization support system that is not dependent on GPS signals.

Table 4.8: Comparison of the GPS and generated coordinate in Scenario 3

Business Name (label)	GPS Coordinate	Generated Coordinate	Verification with Google Map	Distance Difference (meter)
Atasun Optik	lat:40.989670 long:28.780326	lat:40.9900175 long:28.7800944	Verified	43.25
ERCAN BURGER	lat:40.989714 long:28.780182	lat:40.96992640000001 long:28.7804342	Verified	31.72
GLORIA PERFUME	lat:40.989914 long:28.780121	lat:40.9900573 long:28.780811	Verified	16.28
Metin	lat:40.989959 long:28.779832	lat:40.9900289 long:28.7801479	Verified	27.63
Tavuk Dünyası	lat:40.990104 long:28.780083	lat:40.9900417 long:28.7802054	Verified	12.32
Burger King	lat:40.990328 long:28.779948	lat:40.9904457 long:28.7799321	Verified	13.16
Watsons	lat:40.991607 long:28.782118	lat:40.9906355 long:28.779864	Verified	217.8

As mentioned before, we obtain coordinates from GPS and the proposed system during this scenario. However, due to environmental factors such as tall buildings, we lost the GPS signal in some regions while the robot was in motion. In addition, when tracking

the robot's movement on the developed map, it was observed that the sensitivity of the GPS signal decreased. The red point updated itself over the buildings, whereas it should have been in the middle of the road, considering the robot's actual position. As depicted in Figure 4.10, specifically under options a), b), c), d), and e), the red point is situated above the buildings, while the actual position of the robot is in the middle of the street "Barbaros.". In addition, the loss of the GPS signals is demonstrated in Figure 4.10 under options f) and g). The red point is not visible on the map due to the predefined zoom level, in which just a specific area of the blue point is displayed on the map. When the GPS signal is lost, the module returns coordinates with a value of "00.0000". Consequently, the red point is localized in the middle of the Atlantic Ocean. On the other hand, while the robot is moving in the street "Barbaros" from Figure 4.8 from option a) to h), it can be seen that the blue point is updating itself according to business places near the street, except for option b). Here, the coordinate was expected to be near the street and in front of the building; however, the system generated the coordinate on the back side of the building. In general, updating its position near the street and in accordance with the business locations demonstrates the effectiveness of the proposed system in scenarios where GPS signals are either not sensitive or completely lost.

Figure 4.10: Display of GPS coordinate and generated coordinate on Mapbox during Scenario 3

Scenario 4

This scenario was conducted during midday, under clear sunny weather conditions. The experiment started at 12:53 AM and ended at 12:58 AM. As in the second scenario, we could not get a signal from the GPS module due to environmental factors. Therefore, in this scenario, as the mobile robot is in motion, it updates its location on the map solely with the coordinates obtained from the proposed system, which is denoted with a blue point. We manually define a coordinate for the red point near the last business of the scenario to measure the difference with the blue point, in other words, with the generated coordinate. The starting and end points of the robot are opposite to the scenario 3.

In contrast to scenario 3, here, the enhanced system easily detected and recognized dozens of times the business names "watson" as well as "Metin" as shown in Figure 4.12 under options a) and d). When the robot was in front of the business "Tavuk Dünyası," some vibrations occurred due to the cobblestone surface of that region. Thus, the camera angle was altered, but the gimble quickly readjusted it to its standard position. Even in the challenging angles, the system recognizes the name "Tavuk Dünyası," as can be seen under option c). The system recognizes the rest of the businesses without any remarkable challenges. Some frames during the recognition of the names "watson", "Burger King", "GLORIA PARFUME", "ERCAN BURGER" and "Atasun Optik" are demonstrated in Figure 4.12.

In table 4.9, each recognized and corrected business name is shown with their similarity rate. The Easy OCR algorithm recognizes each business name except "Tavuk Dünyası" and "Metin". The behind this was the vibration that occurred during the robot's movement near the business "Tavuk Dünyası". In addition, due to the limitations of the Easy OCR algorithm for handwritten characters, it could not recognize the character "M" when it comes to business "Metin".

Figure 4.11: Detected and recognized business names by the proposed system during Scenario 4.

When table 4.10 was evaluated, it is clear that the distance difference with the red point decreases as the mobile robot approaches the endpoint. However, when the robot approaches the last two businesses that are placed in two adjacent buildings, the distance error does not decrease. This is due to the inaccurate coordinate placement on Google

Table 4.9: Recognized and Corrected Labels with the similarity rate in Scenario 4

Business Name	Recognized Label	Corrected Label with Sequence Matcher	Word Similarity Rate (%)
Watsons	watsons	watsons	100
BURGER KING	KING	KING	100
Tavuk Dünyası	Tavul	Tavuk	80
Metin	Ketin	Metin	80
GLORIA PERFUME	GLORIA PERFUME	GLORIA PERFUME	100
Ercan Burger	ERCAN	ERCAN	100
Atasun Optik	Atasun	Atasun	100

Maps. This shows the limitations of the proposed system. If inaccurate coordinates are placed on Google Maps, generating the highly sensitive coordinates becomes challenging.

As mentioned before, the red point in this scenario was set to a constant position near the last business, "Atasun Optik," to measure the generated system's effectiveness when the mobile robot reaches the endpoint. Examining Figure 4.12 demonstrates that as the robot moves and recognizes business names, the blue point updates its position in accordance with the business places. However, when it recognizes the name "Ercan Burger" the generated coordinate is not located near the road or in front of the building; instead, it is situated on the back side of the building, as can be seen under option f). This discrepancy is due to an inaccurate coordinate placement on Google Maps. The same discrepancy arises with the name "Atasun Optik"; the generated coordinate by the proposed system is not positioned in front of the business "Atasun Optik" but rather in front of a neighboring building, as seen in option g).

Table 4.10: Comparison of the GPS and generated coordinate in Scenario 4

Business Name (label)	GPS Coordinate	Generated Coordinate	Verification with Google Map	Distance Difference (meter)
Watsons	lat:40.98985981445177 long:28.78013560421319	lat:40.9906355 long:28.779864	verified	89.21
Burger King		lat:48.9964457 long:28.7799321	verified	67.35
Tavuk Dünyası		lat:40.9906417 long:28.7802054	verified	21.06
Metin Kuyumculuk		lat:40.9900289 long:28.7801479	verified	18.83
GLORIA PERFUME		lat:40.9900573 long:28.7800811	verified	14.43
ERCAN BURGER		lat:40.98992640000001 long:28.7804342	verified	26.13
Atasun Optik		lat:40.9900175 long:28.7800944	verified	17.87

Figure 4.12: Display of GPS coordinate and generated coordinate on Mapbox during Scenario 4.

4.2.3 Localization in Commercial District

The final experiment was conducted in a commercial district in test Area 3. As shown in Figure 4.13, we designated starting and ending points for the motion of the mobile robot. The test Area 3 has a length of about 90 meters. One side of test Area 3 is bor-

dered by 7 or 8-story buildings, while on the opposite side, there is a road, followed by another set of multi-story buildings. The numbers in Figure 4.13 represent the businesses that are referred to in table 4.11. Similar to the preceding testing areas, each business in this location features complex signboards presented in various fonts and set against diverse backgrounds.

Scenario 5

The experiment was conducted in the evening, under challenging conditions such as low light and a strong storm. To evaluate the performance of the proposed system in challenging conditions, including low light, a strong storm, and cobblestone terrain, the experiment was deliberately conducted during the evening hours on a surface with cobblestones. The robot's initial position is at the start of the business building identified as number one, while its final destination is the building marked with the number five in Figure 4.13. In this experiment, we acquire coordinates from the GPS module and the proposed system. The experiments started at 05:34 AM and ended at 05:33 AM.

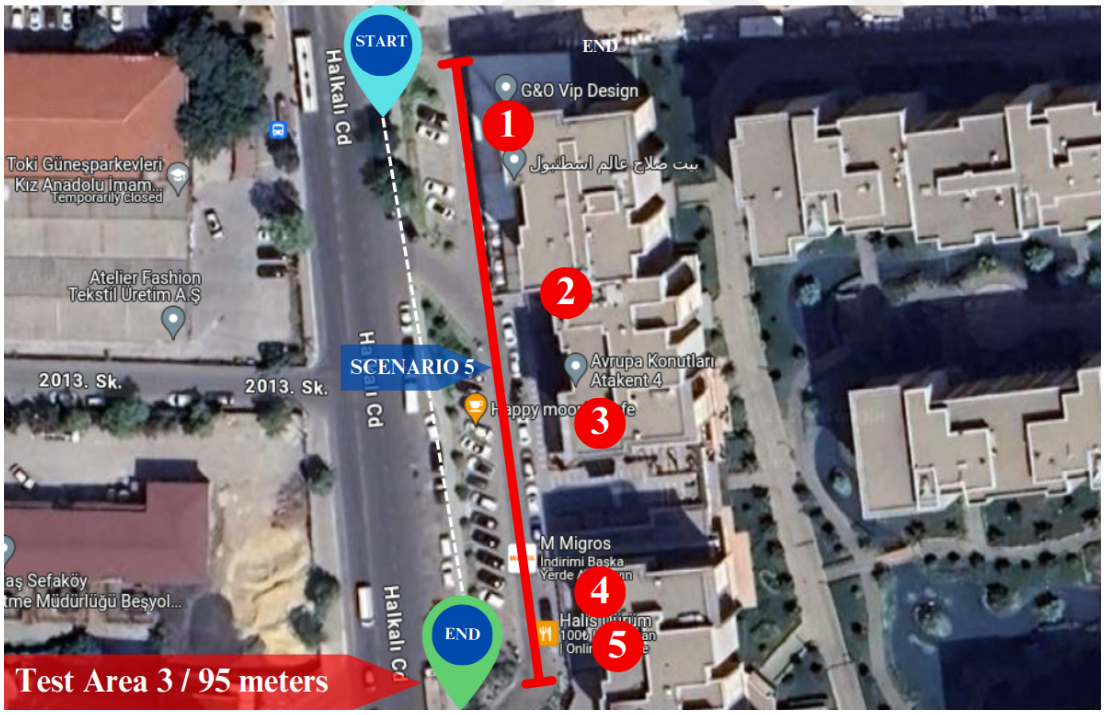


Figure 4.13: Start and End points of the robot for Test Area 3.

In this experiment, the performance of the proposed system for text detection and recognition was robust despite challenges such as low light conditions, surface-induced vi-

Table 4.11: Businesses and corresponding addresses in Test Area 3.

Number	Bussiness Name	Address
1	G&O VIP	Halkalı Caddesi,Atakent 4 No:206L 34290 Küçükçekmece/İstanbul
2	Maydonoz	Halkalı Caddesi,Atakent 4 No:206L 34290 Küçükçekmece/İstanbul
3	Starbucks	Halkalı Caddesi, No:206 34290 Küçükçekmece/İstanbul
4	Lotus Eczane	Halkalı Caddesi, No:206, D:E 34290 Küçükçekmece/İstanbu
5	Halis Dürüm	Avrupa Konutları 4, İnönü Mah. No:206 34290 Küçükçekmece/İstanbul

brations leading to video stream blurriness, and the presence of a strong storm. The system effectively recognized all businesses within the test area except one name. Each recognized business names are demonstrated in Figure 4.14 under options a), b), d), and e). However, under option c), the proposed system encountered difficulty in recognizing the business "STARBUCKS". The reason behind this is the white light on the signboard name and blurriness caused by the storm, which altered the camera position. The reflection of the lights resulted in the characters appearing connected without clear spacing, posing a challenge for the system in recognizing the business name. On the other hand, the strong storm intermittently shifted the camera position throughout the experiments. Fortunately, the gimbal played a crucial role in restoring stability to the camera. The proposed system generally demonstrated remarkable performance in challenging conditions, including low light, a strong storm, and a cobblestone surface.

As indicated in Table 4.12, the EasyOCR algorithm accurately recognizes "VIP", "Maydonoz" and "LOTUS" without missing any characters. However, it erroneously recognizes the name "HALIS" as "nHALIS" Subsequently, this misrecognizing was corrected by the Sequence Matcher algorithm.

Table 4.12: Recognized and Corrected Labels with the similarity rate in Scenario 5

Business Name	Recognized Label	Corrected Label with Sequence Matcher	Word Similarity Rate (%)
G&O VIP	VIP	VIP	100
Maydonoz	Maydonoz	Maydonoz	100
STARBUCKS	-	-	-
LOTUS	LOTUS	LOTUS	100
HALIS	nHALIS	HALIS	87.5

a) b) c) d) e)

Figure 4.14: Detected and recognized business names by proposed system during Scenario 5.

Table 4.13 compares the coordinates from the GPS module with the generated coordinates, emphasizing the differences in distance between the two sets of coordinates. Initially, when the robot was in front of business number 1, the distance difference was approximately 18 meters. Subsequently, as GPS signals were lost, the distance difference increased to thousands of kilometers. At one point during the test when GPS signals briefly returned, the distance difference at that position was around 15 meters, although this occurrence was short-lived.

As stated at the beginning of this scenario, during this experiment, while the mobile robot moving in defined test area, we are acquiring coordinates from both the GPS and the proposed system. As can be seen in Figure 4.15 under option a), when mobile robot

Table 4.13: Comparison of the GPS and generated coordinate in Scenario 5.

Business Name (label)	GPS Coordinate	Generated Coordinate	Verification with Google Map	Distance Difference (kms)
G&O VIP	lat:41.021457 long:28.791342	lat:41.021316 long:28.791462	verified	0.018
Maydonoz	lat:0.0000 long:0.0000	lat:41.02093 long:28.791677	verified	5,405.025
STARBUCKS	lat:0.0000 long:0.0000	-	-	-
Lotus Eczane	lat:0.0000 long:0.0000	lat:41.0205738 long:28.7916786	verified	5,404.995
Halis Dürüm	lat:0.0000 long:0.0000	lat:41.020566 long:28.791589	verified	5,404.989

was in front of the business “G&O VIP” the blue point accurately positioned itself in the front of the building, whereas the GPS coordinate, represented by the red point, was situated in the side street. Subsequently, when the mobile robot reached the business marked with number 2 in Table 4.11, the GPS signals were lost. By default, the module provided default coordinates, such as (0.0000, 0.0000). Consequently, the red point positioned itself near the intersection of Ecuador (latitude 0 degrees), as depicted in option b). After the initial loss of GPS, there was a brief period during which GPS signals was available, but then signals was lost once again, and we were unable acquire data for the rest of the test. However, the robot’s localization persisted using the generated coordinates from the proposed system until the end of the test. As evident in options c), d), and e), the blue point consistently updated its position as the business names were recognized. This underscores the significance of developing a localization system that does not solely rely on GPS signals in urban areas.

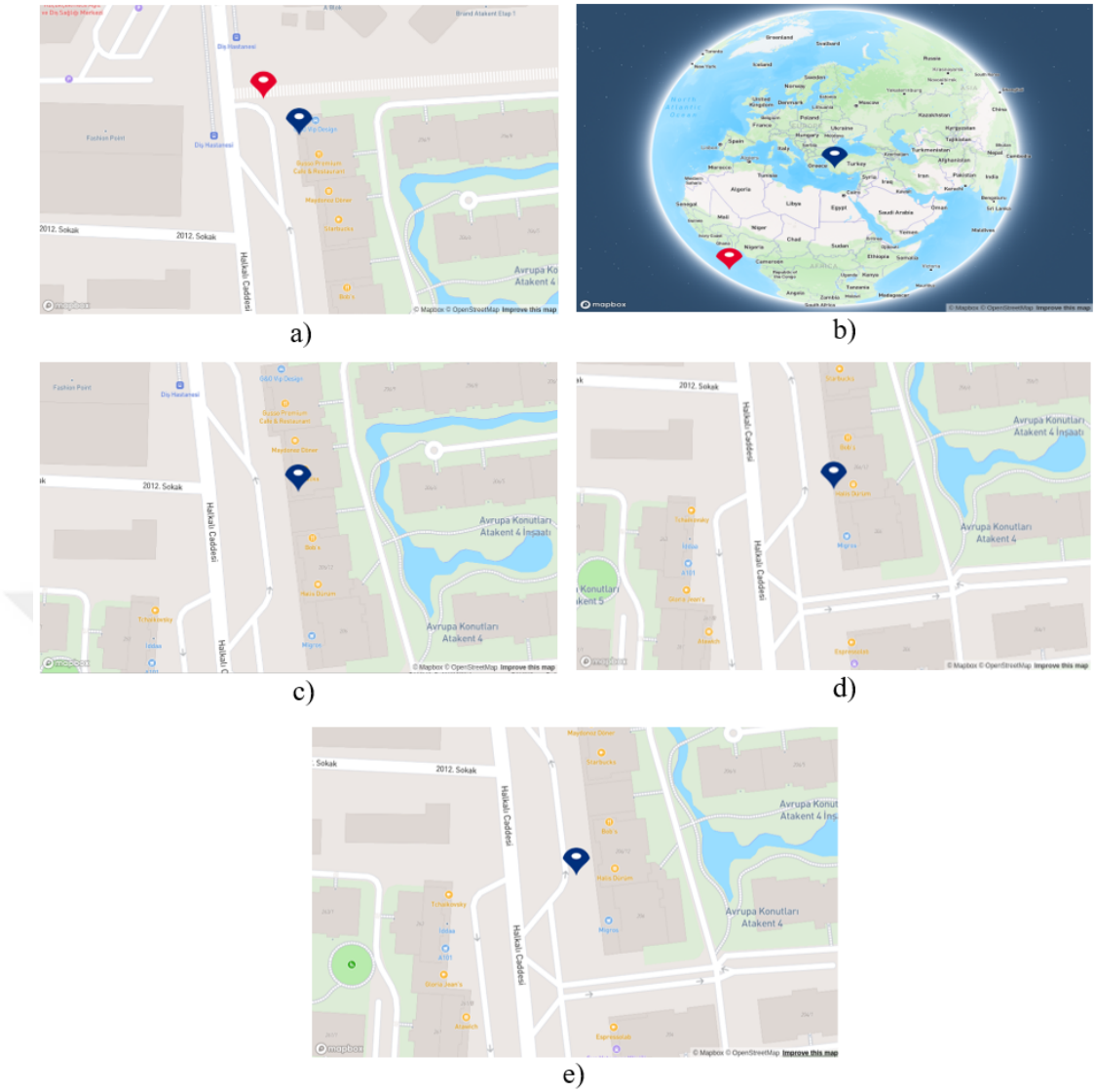


Figure 4.15: Display of GPS coordinate and generated coordinate on Mapbox during Scenario 5.

4.2.4 System Evaluation

In this dissertation, a visual based localization system was developed. When similar studies are examined in Table 4.14, most of the studies use few sensors in order to localize mobile robots and the distance error between the reference and generated coordinates are given as minimum, maximum and average. In (Nilwong et al., 2019) the three different sensors are utilized. Even so, the average error rate was around 28 m. While it is crucial to conduct localization tests under various scenarios and in real time, (Correa & Soto, 2010) carried out their experiments through simulation and (Vishal et al., 2015) performed their experiments via the video recordings. Despite they achieved

error rate 4.23 m and 7.5 m, their studies lack of real time localization in urban environments. By employing three different sensors, (Cai et al., 2019) achieved an average error rate of 4.12 meters. While this is a closely comparable error rate to our work, it is noteworthy that we attained a marginally higher error rate of 4.20 meters using only one sensor. Considering both the average and minimum error rates in our system, it is lower than the other studies. Consequently, the high accuracy and sensitivity of the proposed system was demonstrated through real-time experiments conducted in various test areas and light conditions. Taking into account both the average and minimum error rates in our system, they are lower than those presented in Table 4.14, except for the study by (Cai et al., 2019), where they achieved slightly lower error rates by a few centimeters. In general, the proposed system demonstrated high accurate localization through real-time experiments conducted in various test areas and light conditions.

Table 4.14: Comparison of the proposed model and existing studies

Research	Sensors	Method	Environment	Experiment Type	Error Rate (meters)
(Nilwong et al., 2019)	GPS/Compass/ Camera	Faster R-CNN	Outdoor	Real-Time	Average:28 Minimum:1
(Cai et al., 2019)	GPS, IMU, Visual Odometry	Extended Kalman Filter	Outdoor	Real-Time	Average:4
(Vishal et al., 2015)	GPS/ Contour Action Camera	Image Retrieval Algortihm	Outdoor	Video Record	Average:7
(Correa & Soto, 2010)	Camera	Bayesian non-parametric estimation & partical filter	Outdoor	Simulation	Average:4
(Chiang & Huang, 2008)	INS/GPS	Kalman Filter	Outdoor	Real-Time	Average:16
Proposed	**GPS/Camera**	**EAST+EASYOCR+SM & WEB MINING**	**Outdoor**	**Real-Time**	**Average:4 Minimum:1**

In Table 4.15, we evaluated the operational efficiency of the proposed system, emphasizing its processing speed and average Frames Per Second (FPS). The processing speed refers to the duration elapsed from text detection to the reflection of coordinates on the map. The proposed system exhibited an average processing speed of approximately 2 seconds. Given the significance of time in the context of mobile robot localization,

Table 4.15: Performance evaluation of proposed system in terms of FPS and Processing Time

Method	Scenario	Average FPS	Average Time (seconds)
Proposed System	1	9.10	2.93
	2	7.55	2.60
	3	5.90	2.39
	4	5.10	2.10
	5	8.50	2.46

this performance can be considered as remarkable. On the other hand, the proposed system, running on a laptop with limited computational capacity, achieved an average of 8 FPS.

CHAPTER FIVE
CONCLUSION

This dissertation introduces a visual localization system designed for use in urban environments where GPS signals are either low or blocked. The proposed system consists of three steps: 1) Text detection-recognition, 2) generating coordinates, and 3) development of a map. For the first step, a text detection and recognition system was proposed. The EAST text detection algorithm was applied in order to detect the texts in the urban environments. To enhance the performance of the text detection algorithm we perform post-processing. Following, for recognition of the detected text Easy OCR, Keras OCR, and Tesseract OCR algorithms were employed, and a comparative analysis was conducted. In order to recognize the detected text, the algorithms Easy OCR, Keras OCR, and Tesseract OCR were employed. In addition, a word correction algorithm was applied to correct misrecognized words. Then a comparative analysis was conducted to determine the best-performing algorithm. In the second step, recognized labels were fed to Places API in order to obtain POI data, especially coordinates, for a particular area. Through this process, POI data which includes the names of businesses, their coordinates in terms of longitude and latitude format, operational status, and so on was collected for specific areas. Moreover, the algorithm was developed which will search in the obtained data for the relevant label and return the most suitable coordinate using the Haversine formula in scenarios where GPS signals are blocked. In the last step, a map was developed by utilizing the Mapbox tool. The coordinates obtained from the GPS module and those generated by the proposed system are represented on the map with red and blue points. The red points demonstrate the coordinates of GPS, while the blue point demonstrates the generated coordinates from the proposed system. Moreover, to evaluate the error rate of the proposed system, the distance between the generated coordinate and the GPS module coordinate is calculated by utilizing the Haversine Formula. The experiments carried out for text detection and recognition show that enhancing the system by applying post-processing and word similarity improves the recognition accuracy of the OCR algorithms. Additionally, the proposed system's performance for localization with a delivery robot was evaluated in three different areas. Effective text detection and recognition performance, obtaining precise

POI data, and developing an algorithm that returns respective coordinates contributed to the effective localization performance of the proposed system, resulting in an approximate error of around 4 meters.

In future work, we aim to eliminate blurriness in the stream caused by surface vibrations by utilizing image processing filters and computer vision techniques. Furthermore, we aim to decrease the error rate in the localization task of the proposed system. This will be achieved by determining the pose of the robot and measuring the distance between the camera and the respective business name. Therefore, the localization of the robot will be determined more accurately.

REFERENCES

Al Khatib, E. I., Jaradat, M. A. K., & Abdel-Hafez, M. F. (2020). Low-cost reduced navigation system for mobile robot in indoor/outdoor environments. IEEE Access, 8, , 25014–25026.

Antoniou, V., Morley, J., & Haklay, M. (2009). Tiled vectors: A method for vector transmission over the web. In Web and wireless geographical information systems: 9th international symposium, w2gis 2009, maynooth, ireland, december 7-8, 2009. proceedings 9 (pp. 56–71).

Baek, Y., Lee, B., & Han. (2019). Character region awareness for text detection. In Proceedings of the ieee/cvf conference on computer vision and pattern recognition (pp. 9365–9374).

Bansal, A., Badino, H., & Huber, D. (2014). Understanding how camera configuration and environmental conditions affect appearance-based localization. In 2014 ieee intelligent vehicles symposium proceedings (pp. 800–807).

Cabon, Y., Murray, N., & Humenberger, M. (2020). Virtual kitti 2.

Cai, G.-S., Lin, H.-Y., & Kao, S.-F. (2019). Mobile robot localization using gps, imu and visual odometry. In 2019 international automatic control conference (cacs) (pp. 1–6).

Chen, N., Wang, H., Fan, G., Yang, D., Rao, L., et al. (2023). An end-to-end robotic visual localization algorithm based on deep learning. Hindawi.

Chiang, K.-W., & Huang, Y.-W. (2008). An intelligent navigator for seamless ins/gps integrated land vehicle navigation applications. Applied Soft Computing, 8, 1, 722–733.

Conceição, T., Neves dos Santos, F., & Costa. (2018). Robot localization system in a hard outdoor environment. In Robot 2017: Third iberian robotics conference: Volume 1 (pp. 215–227).

Correa, J., & Soto, A. (2010). Active visual perception for mobile robot localization. Journal of Intelligent and Robotic Systems, 58, , 339–354.

Difflib. (2023). Difflib Library Documentation. https://docs.python.org/3/library/difflib.html. ((Accessed: 10.04.2023))

Epshtein, B., Ofek, E., & Wexler, Y. (2010). Detecting text in natural scenes with stroke width transform. In 2010 ieee computer society conference on computer vision and pattern recognition (pp. 2963–2970).

Fleet, D., Pajdla, T., & Schiele. (2014). Computer vision–eccv 2014: 13th european

conference, zurich, switzerland, september 6-12, 2014, proceedings, part i (Vol. 8689). Springer.

Gaidon, A., Wang, Q., & Cabon. (2016). Virtual worlds as proxy for multi-object tracking analysis. In Cvpr.

Garg, S. (2020, September). Nordland-part-2020. Zenodo. Retrieved from https://doi.org/10.5281/zenodo.4016653 doi: 10.5281/zenodo.4016653

Georgiev, A., & Allen, P. K. (2004). Localization methods for a mobile robot in urban environments. IEEE Transactions on Robotics, 20, 5, 851–864.

Graves, A., Fernández, S., & Gomez. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd international conference on machine learning (pp. 369–376).

He, K., Zhang, X., & Ren. (2016). Deep residual learning for image recognition. In Proceedings of the ieee conference on computer vision and pattern recognition (pp. 770–778).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9, 8, 1735–1780.

Inman, J. (1849). Navigation and nautical astronomy: For the use of british seamen. F. and J. Rivington.

Jiang, S., Song, W., & Zhou. (2022). Stability analysis of the food delivery robot with suspension damping structure. Heliyon, 8, 12, .

Kamarudin, S. S., & Tahar, K. N. (2016). Assessment on uav onboard positioning in ground control point establishment. In 2016 ieee 12th international colloquium on signal processing & its applications (cspa) (pp. 210–215).

Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., … others (2015). Icdar 2015 competition on robust reading. In 2015 13th international conference on document analysis and recognition (icdar) (pp. 1156–1160).

Kaya, O. M. T., & Erdemir, G. (2023). Design of an eight-wheeled mobile delivery robot and its climbing simulations. In Southeastcon 2023 (pp. 895–900).

Kendall, A., Grimes, M., & Cipolla, R. (2015). Posenet: A convolutional network for real-time 6-dof camera relocalization. In Proceedings of the ieee international conference on computer vision (pp. 2938–2946).

KerasOCR. (2023). Keras OCR Documentation. https://keras-ocr.readthedocs.io/en/latest/. ((Accessed: 01.04.2023))

Khaki, S., Pham, H., Han, Y., Kuhl, A., Kent, W., & Wang, L. (2020). Convolutional

neural networks for image-based corn kernel detection and counting. Sensors, 20, 9, 2721.

Kümmerle, R., Ruhnke, M., & Steder. (2013). A navigation system for robots operating in crowded urban environments. In 2013 ieee international conference on robotics and automation (pp. 3225–3232).

Lee, A. J., Song, W., Yu, B., Choi, D., Tirtawardhana, C., & Myung, H. (2023). Survey of robotics technologies for civil infrastructure inspection. Journal of Infrastructure Intelligence and Resilience, 2, 1, 100018.

Liao, M., Shi, B., & Bai, X. (2018). Textboxes++: A single-shot oriented scene text detector. IEEE transactions on image processing, 27, 8, 3676–3690.

Maddern, W., Pascoe, G., & Linegar. (2017). 1 year, 1000 km: The oxford robotcar dataset. The International Journal of Robotics Research, 36, 1, 3–15.

Mapbox. (2023a). Mapbox Documentation. https://docs.mapbox.com/maps. ((Accessed: 7.11.2023))

Mapbox. (2023b). Mapbox Documentation. https://docs.mapbox.com/api/maps/styles/. ((Accessed: 7.11.2023))

Meddeb, H., Abdellaoui, Z., & Houaidi, F. (2023). Development of surveillance robot based on face recognition using raspberry-pi and iot. Microprocessors and Microsystems, 96, , 104728.

Naseer, T., Burgard, W., & Stachniss, C. (2018). Robust visual localization across seasons. IEEE Transactions on Robotics, 34, 2, 289–302.

NearbySearch. (2023). Places API Documentation. https://developers.google.com/maps/documentation/places/web-service?hl=tr. ((Accessed: 10.11.2023))

Netek, R., Masopust, J., & Pavlicek. (2020). Performance testing on vector vs. raster map tiles—comparative study on load metrics. ISPRS International Journal of Geo-Information, 9, 2, 101.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., & Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning.

Nilwong, S., Hossain, D., & Kaneko. (2019). Deep learning-based landmark detection for mobile robot outdoor localization. Machines, 7, 2, 25.

Noskov, A. (2018). Computer vision approaches for big geo-spatial data: quality assessment of raster tiled web maps for smart city solutions. In 7th international conference on cartography and gis; bulgarian cartographic association: Sofia, bulgaria (pp. 296–305).

Piasco, N., Sidibé, D., & Gouet-Brunet. (2021). Improving image description with auxiliary modality for visual localization in challenging conditions. International Journal of Computer Vision, 129, , 185–202.

Pixhawk. (2023). Pixhawk Documentation. https://pixhawk.org/. ((Accessed: 02.10.2023))

Qiao, Y., Cappelle, C., & Ruichek, Y. (2017). Visual localization across seasons using sequence matching based on multi-feature combination. Sensors, 17, 11, 2442.

Quectel. (2023). Quectel Documentation. https://www.quectel.com/ProductDownload/. ((Accessed: 11.11.2023))

Ranjbarzadeh, R., Jafarzadeh Ghoushchi, S., Anari, S., Safavi, S., Tataei Sarshar, N., Babaee Tirkolaee, E., & Bendechache, M. (2022). A deep learning approach for robust, multi-oriented, and curved text detection. Cognitive computation, , , 1–13.

Ratcliff, J. W., Metzener, D., et al. (1988). Pattern matching: The gestalt approach. Dr. Dobb's Journal, 13, 7, 46.

Ravankar, A., Ravankar, A. A., & Kobayashi. (2018). Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges. Sensors, 18, 9, 3170.

Roshan, A., Ardeshir, S., & Shah, M. (2014). Gps-tag refinement using random walks with an adaptive damping factor. In Proceedings of the ieee conference on computer vision and pattern recognition (pp. 4280–4287).

Shi, B., Bai, X., & Yao, C. (2016). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE transactions on pattern analysis and machine intelligence, 39, 11, 2298–2304.

Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., & Fitzgibbon, A. (2013). Scene coordinate regression forests for camera relocalization in rgb-d images. In Proceedings of the ieee conference on computer vision and pattern recognition (pp. 2930–2937).

Sinha, H., Patrikar, J., & Dhekane. (2018). Convolutional neural network based sensors for mobile robot relocalization. In 2018 23rd international conference on methods & models in automation & robotics (mmar) (pp. 774–779).

Sinnott, R. (1984). Virtues of the haversine, vol. 68. Sky and telescope, , , 158.

Smith, R. (2007). An overview of the tesseract ocr engine. In Ninth international conference on document analysis and recognition (icdar 2007) (Vol. 2, pp. 629–633).

Stereolabs. (2023). Zed2i Documentation. https://www.stereolabs.com/docs/. ((Accessed: 02.10.2023))

Sturm, J., Engelhard, N., & Endres, F. (2012, Oct.). A benchmark for the evaluation of rgb-d slam systems. In Proc. of the international conference on intelligent robot systems (iros).

Sun, Hu, Y., & Ma. (2023). Conflating point of interest (poi) data: A systematic review of matching methods. Computers, Environment and Urban Systems, 103, , 101977.

Sun, R., Wang, G., & Zhang. (2020). A gradient boosting decision tree based gps signal reception classification algorithm. Applied Soft Computing, 86, , 105942.

Ublox. (2023). Ublox Documentation. https://www.u-blox.com/en/product/neo-m8-series. ((Accessed: 12.11.2023))

UsaGovernment. (2023). Technical Documentation. https://www.gps.gov/systems/gps/performance/accuracy/. ((Accessed: 02.11.2023))

Vishal, K., Jawahar, C., & Chari, V. (2015). Accurate localization by fusing images and gps signals. In Proceedings of the ieee conference on computer vision and pattern recognition workshops (pp. 17–24).

Wu, J., Shi, Q., Lu, Q., Liu, X., Zhu, X., & Lin, Z. (2022). Learning invariant semantic representation for long-term robust visual localization. Engineering Applications of Artificial Intelligence, 111, , 104793.

Yang, C., Goodchild, M., Huang, Q., Nebert, D., Raskin, R., Xu, Y., … Fay, D. (2011). Spatial cloud computing: how can the geospatial sciences use and help shape cloud computing? International Journal of Digital Earth, 4, 4, 305–329.

Yao, C., Bai, X., & Liu. (2012). Detecting texts of arbitrary orientations in natural images. In 2012 ieee conference on computer vision and pattern recognition (pp. 1083–1090).

Yasuda, Y. D., Martins, L. E. G., & Cappabianco, F. A. (2020). Autonomous visual navigation for mobile robots: A systematic literature review. ACM Computing Surveys (CSUR), 53, 1, 1–34.

Yousef, M., Hussain, K. F., & Mohammed, U. S. (2020). Accurate, data-efficient, unconstrained text recognition with convolutional neural networks. Pattern Recognition, 108, , 107482.

Yousuf, S., & Kadri, M. B. (2018). Robot localization in indoor and outdoor environments by multi-sensor fusion.. (In 2018 14th International Conference on

Emerging Technologies (ICET), pp. 1–6)

Yu, Y., Zhang, K., & Liu. (2020). Real-time visual localization of the picking points for a ridge-planting strawberry harvesting robot. IEEE Access, 8, , 116556–116568.

Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., & Liang, J. (2017). East: an efficient and accurate scene text detector. In Proceedings of the ieee conference on computer vision and pattern recognition (pp. 5551–5560).

Zuo, L.-Q., Sun, H.-M., & Mao. (2019). Natural scene text recognition based on encoder-decoder framework. IEEE Access, 7, , 62616–62623.

BIOGRAPHY

Erdal ALİMOVSKİ received his B.S degree in Informatics and Computer Engineering from St. Clement of Ohrid University, Bitola/North Macedonia, and M.S degree in Computer Engineering from Istanbul Sabahattin Zaim University Istanbul/Turkey, in 2014 and 2019 respectively. Currently, he is enrolled as a Ph.D. candidate in Department of Computer Engineering in Istanbul Sabahattin Zaim University, where he is also rendering his services as research and teaching assistant. His research interests include computer vision, deep learning, web mining, robotics, and generative models.