INDOOR NAVIGATION FOR THE VISUALLY IMPAIRED BY READING DOOR LABEL

BY

KUAN WEI YEOW

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)
Faculty of Information and Communication Technology
(Kampar Campus)

JAN 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: ___Indoor Navigation for the visually impaired by reading door label __

Academic Session: ___JAN 2024__

I ____________________KUAN WEI YEOW______________________

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)

(Supervisor's signature)

Address:

_No.96, Jalan Pelanduk 3,______

_Taman Suntex, 43200,________

_Cheras, Selangor_____________

Leung Kar Hang

Supervisor's name

Date: _25 April 2024_________

Date: 25 April 2024

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITY TUNKU ABDUL RAHMAN

Date: __25 April 2024_______

SUBMISSION OF FINAL YEAR PROJECT

It is hereby certified that ____Kuan Wei Yeow______ (ID No: __20ACB03477___) has completed this final year project entitled "__Indoor Navigation for the Visually Impaired by Reading Door Label__" under the supervision of __Prof. Dr Leung Kar Hang__ (Supervisor) from the Department of ___Computer Science, Faculty of Information and Communication Technology__ .

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

____

(*KUAN WEI YEOW*)

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Indoor Navigation for the Visually Impaired by Reading Door Label**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ______________________________

Name : ____KUAN WEI YEOW_____

Date : ___25 April 2024___________

ACKNOWLEDGEMENTS

I would like to express thanks and appreciation to my supervisor, Prof. Dr Leung Kar Hang and my moderator, Ts Dr Tan Hung Khoon who have given me a golden opportunity to involve in the Image Processing field study. Besides that, they have given me a lot of guidance in order to complete this project. When I was facing problems in this project, the advice from them always assists me in overcoming the problems. Again, a million thanks to my supervisor and moderator.

Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

Recently, there have been many efforts from people implementing technology that can provide help for people with special needs. Similarly, the attention and study on the indoor navigation for the visually impaired was getting more attention from the world. The aim of these is to further enhance the quality of daily life of the visually impaired by enabling them to travel alone. The major challenge of this research is how to provide correct navigation to the visually impaired by utilizing the information about their surroundings in an indoor environment. As a general knowledge, signs, and label was widely used in the indoor environment to navigate peoples the direction of the locations. Unfortunately, the visually impaired was unable to see those signs and label due to their disability. Therefore, in this project, an indoor navigation system was implemented by reading door labels. The system only used a laptop, camera, and earphones to perform the navigation. The captured images of the user's surroundings will be sent to the laptop for further processing. A pre-recorded map of the indoor environment will be installed in the system to perform shortest path calculation or current location notification when the door label is detected. Lastly, the navigation instructions will be converted into speech to convey to the user. The system was controlled in a low-cost implementation since not all of the visually impaired are able to afford high specification of navigation hardware. Therefore, the limitations of this project might be related to the quality of the image acquired and the processing power of the laptop. Both issues were related to the performance of the system, and these might affect the accuracy of navigation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
RFID	Radio Frequency Identification
GPS	Global Positioning System
AR	Augmented Reality
AI	Artificial Intelligent
MSVI	Moderate to Severe Visual Impairment
IMU	Inertial Measurement Unit
OST	Optical See-Through
BA	Bundle Adjustment
LIDAR	Light Detection and Ranging
LASS	Assist Spatial Sensing
HRTF	Head-Related Transfer Function
OCR	Optimal Character Recognition
CNN	Convolutional Neural Network
DSP	Digital Signal Processor
DETR	Detection Transformer
FFN	Feedforward Network
TTS	Text-to-Speech
TWSI	Tactile Walking Surface Indicator
BLE	Bluetooth Low Energy
IDR	Inertial Dead Reckoning
gTTS	Google Text-to-Speech
FICT	Faculty of Information and Communication Technology
UTAR	University Tunku Abdul Rahman

CHAPTER 1

Introduction

1.1 Problem Statement and Motivation

When normal people travel in a new environment, the common way for them to locate themselves is by road signs, door labels, landmarks, maps, and trademarks. Unfortunately, these pieces of information can only be noticed by people through their vision. This made the visually impaired face the difficulty of locating and navigating themselves when traveling alone. Even though there was Braille signage to help navigate the visually impaired, they may also be suffering about where was the Braille signage and how to reach their destination. This shows that external help is still needed for them to navigate themselves to their destination safely. The problem the visually impaired faced was how to continuously receive visual information so they could know their current location all the time.

Besides that, path navigation is important to the visually impaired since they have no idea about the structure of the buildings. The distance and direction of their destination were unknown to them while traveling. For example, it is hard for them to decide to turn left or right at the end of the corridor. Even when we explained the structure of the building to them, it was still hard for them to memorize and navigate correctly in a complex structure of buildings. The problem the visually impaired faced was how they navigated themselves to reach their destination in the shortest path. Therefore, we need a technology that can continuously guide them accurately from a starting point to their destination.

Accuracy is crucial in indoor navigation systems since wrong detection or misdetection will confuse the visually impaired when they are navigating for a location. For example, the visually impaired could get lost inside the building when the system accuracy is low and cannot detect some door labels or signage. Most indoor navigation systems by using a camera could be facing the problem of capturing blurred images due to the motion of the visually impaired when they were walking around. Besides that, the noise from the background such as the poster will annoy the reading of the door label. This was the critical reason to explain the low accuracy of the indoor navigation system. The main problem was how to increase the accuracy and trustworthiness of the indoor navigation system by reading door labels.

1.2 Project Scope

The thesis aims to propose a camera-based indoor navigation system that detects only door labels to guide the user while traveling. This system will continuously inform the user once the door label is detected. If the system wants to navigate the user properly, the information except for the door label is all redundant and will only confuse the user while using the system. Therefore, to focus only on the door label, a red dot will be placed beside the door label to locate the door label precisely. With the help of the red colour dot, it could help to reduce the burden of the system by not running the image-to-text algorithm all the time.

Besides that, a pre-defined map and voice navigation will be embedded in the system. The system will first receive the input from the user, and then start to calculate the nearest route from the user's current location to their destination. With a pre-defined map, the system will be able to guide the user to travel from point A to point B by giving some instructions such as “Go straight”, “turn right”,“ turn left at the end of the corridor”, and “your destination will be at your left-hand side. With the help of voice navigation, the user will hardly miss their destination or turn into the wrong route.

1.3 Project Objectives

Navigation in outdoor environments for the visually impaired faces fewer challenges than indoor navigation because of the advance of the Global Positioning System (GPS). GPS is inapplicable in an indoor environment since that is not under the field of navigation in GPS. The common way for them to navigate themselves is with the assistance of a guide dog or white cane to travel from one point to another. The disadvantage of these methods is also critical as both of those methods are not suitable for a crowded environment or the pet is restricted. Even with the assistance of these methods, it is still not a good navigation tool as orientation and precise navigation cannot be provided to the user and cause the visually impaired to doubt the path of the journey. Therefore, the system proposed will use a pre-defined map to navigate the user precisely to reach their destination.

The project aims to achieve a high accuracy of the door label detection of the system. It will be meaningless if the system cannot detect and read the door label correctly. It might be challenging since the movement of the user might affect the quality of the image captured by the camera for processing. Therefore, the system proposed will try to achieve a steady and clear way to capture the image of the door label on every frame. After that, this project is trying to build the system with a low budget so that everyone is affordable for the system hardware. The only input device used in the system is a webcam and a microphone.

The system will be implemented on a minimum budget so it could be affordable for everyone. A camera, laptop, and earphones are used to support all the functions of the system. Devices such as smart glasses, Kinect depth cameras, and ultrasonic rangefinders are expensive to perform similar tasks and they might not be able to afford it.

1.4 Impact, significance, and contributions

According to [1], 253 million people are estimated to have the issue of visual impairment worldwide. To be more detailed, among the 253 million people there were 217 million had moderate to severe visual impairment (MSVI) and 36 million were blind. Aged above 50 occupy 80% of the visually impaired people worldwide. With the modern technology and knowledge nowadays, tasks to improve visually impaired people's daily lives such as navigation, text-to-speech, and speech-to-text still facing challenges in research and development.

By having this indoor navigation system, the ability to independent movement by the visually impaired inside a building will be improved. The recent technology such as Bluetooth low energy (BLE) is not too suitable for every environment. This is because not every place will install the BLE beacons in their building because of the cost. The proposed system can be run in every indoor environment by just using a camera and a device to run the program. The cost for red colour dot is also cheap to install on every door label in the building.

1.5 Background Information

Navigation is important in our daily lives as we are always navigating for school, restaurants, office, work, and other reasons. Most of the people rely on their vision when moving from one spot to another. This is because most of the navigation information is presented visually to inform the people about where the place is located. Visually impaired would face difficulty in navigating or orienting themselves while traveling around by themselves. Therefore, the field of indoor navigation for the visually impaired focuses on finding solutions to let them navigate themselves in indoor environments such as shopping malls, schools, hospitals, and other public venues. This intention accelerates the development of technology to help those who are visually impaired to travel alone without special care.

The earliest technology such as tactile floor patterns and Braille signage was implemented to convert visual information into tactile information that was able to be received by visually impaired persons. With the rapid improvement of technology over time, the indoor navigation system was applicable with the help of devices such as cameras, Radio Frequency Identification (RFID), Bluetooth beacons and Wi-Fi Positioning, Global Positioning System (GPS), and Augmented Reality (AR) to gather environmental information and transfer it in the form of data. Then, software-based technology such as computer vision, image processing, and artificial intelligence (AI) techniques were developed and contributed to the navigation system by translating those data into voice or haptic information to guide visually impaired individuals. The main objective of an indoor navigation system is to guide the user to their desired destination with a safe and shortest route.

The research and development of indoor navigation by reading door labels will be shown in the following chapters. In Chapter 2, similar research and development of indoor navigation systems will be reviewed. Then, the methodology of the system will be presented in Chapter 3. And then, Chapter 4 is the system design of the proposed system. Furthermore, system implementation will be written in Chapter 5. Chapter 6 will talk about the system evaluation and discussion. The last chapter, Chapter 7 will be the conclusion and recommendation.

CHAPTER 2

Literature Reviews

2.1 Review on Different Technologies for Visually Impaired Navigation

In this article [11], there are many technologies reviewed such as the Visual image system, non-visual data system, Map-based system, System with three-dimensional sound, and Smartphone-based solution. We will only focus on the Visual image system in this article. A visual image system is navigation by a computer vision algorithm and sensors that extract the visual image of the surrounding environment by cameras.

2.1.1 Stereo Camera

[2] developed a navigation system that combines a bicycle helmet with earphones, an inertial measurement unit (IMU), and a binocular camera (Figure 2.1.1). IMU was used to track the head position of the user with minimum latency. Texture information as well as distance measurement was provided by the binocular camera.

Figure 2.1.1 Hardware configuration of a bicycle helmet, binocular camera, and earphones [2]

A geometric scene model is used to structure independently the moveable foreground objects from the background. This is because the background in the scene such as the alignment of facades is always static and provides valuable orientation hints to the user. Therefore, the scene geometric model consists of a composition of planar surfaces in global 3D space [2]. We need to proceed with the least-squares optimization and a combination of multi-model RANSAC to locate these planes because it was a multi-model fitting problem [2]. A non-Euclidean disparity space is used to estimate the Planes before they transform into Euclidean 3D space. Therefore, this plane estimation was needed to regularize by applying a set of geometric constraints. For example, those planes that are orthogonal to a ground plane will be constrained by the set of geometric constraints. Vanishing directions are used in a structured urban environment in which it is aligned with the building and the ground that corresponds to normal vectors of the façade and it is helpful for plane tracking when under some crucial conditions.

Distinguishing the specific foreground object is not the main objective in this paper so we just group them into a unified representation. Therefore, the foreground object will be defined as not belonging to the scene. All the data that disparity from all the points and belongs to the background are removed then the remaining points that are located close to each other are partitioned into small segments. Then, the Single Linkage Agglomerative clustering was applied and used to measure the distance of disparity from one point to another. The segment is grouped into objects by applying a reasoning process based on instantiated objects that developed previously in the environment model. Two features are used to assign the closest object to each segment obtained that overlaps in the image space of a segment with the projected outline of an object and Mahalanobis distance in 3d-space between segment and object. Finally, the data of the object such as the dimensions of the bounding box, estimated velocity, and position will be generated by the Kalman filter. A new object will be created in the environmental model the segments do not belong to any of the existing objects. The object will be deleted when it has not been captured for several continuous frames.

Tracking real-time movement and the posture of the head of the user is a must to give the user a good experience by providing consistent acoustic feedback. There are two ways for head tracking: using IMU or using camera-based visual odometry. Devices such as 3-axis accelerometers can capture the angle of the orientation of the user's head and the rotation rates with the help of magnetic field sensors and gyroscopes then pass them into the IMU for estimation. Unfortunately, the head position and translation estimated by IMU are not reliable. Visual odometry tracks the head position and rotation by tracking the movement of salient points in the image from time to time. If most of the salient points are located on the nondynamic background objects, the ego-motion can be derived easily. The only shortage of this method is it has a delay of 50-100ms. Both of this method is used as an integrated approach to overcome the shortage of each other. The focus part on this approach is on the data from the magnetic field measurements and whether inertial is working well in the common orientation filter or not. The orientation was represented by a quaternion and gyroscope in the filter state. In the setup, all the points ss are always obtained by the IMU measurement while visual odometry measurement is only for t_k,t_{k+n}, and t_{k+2n}…. The clone is augmented in time t_k, and the calculation for visual odometry is only available at the time t_{k+n} which the calculation needed the difference between the cloned orientation and present orientation. This gives an innovative idea for the filter by using one extra step. Besides that, the position of the user's head is also measurable outside of the filter by the visual odometry.

The position of obstacles and objects near the user can be detected by the current head pose based on the environment model. There will be a sound source for each object detected which encodes the spatial location through binaural rendering. It is a technique that creates a sound that can help the user differentiate the distance and direction of the object with the help of headphones.

2.1.2 IP Camera Network

[3] proposed a system in which cameras were installed on the ceiling to track the user. The photos acquired will be analyzed and processed by a server to perform computer vision tasks. Figure 2.1.2 shows the system architecture of the indoor navigation system between remote processing computers and mobile applications. The user needed to hold only his smartphone the whole time during his displacement to enable the receiver to the user's voice command by sending it to the remote processing system in text format through WIFI or Bluetooth communication. Then, the remote process system will guide the user to their destination based on an image analytic algorithm by analyzing the photos from different angles from the cameras.

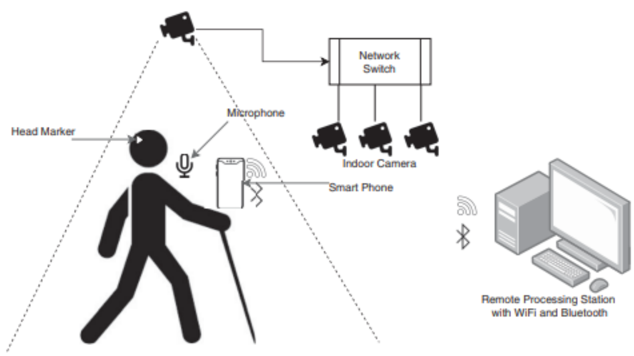


Figure2.1.2 Proposed System Approach Architecture [3]

There are only two main functions that the mobile application consists of that is speech-to-text and text-to-speech converter. First, the application will convert the voice command from the user into text and forward it to the remote process system through a WIFI connection. Then, the image processing system will respond to the application with text that contains navigational information. Finally, the text will be converted into a voice message and guide the user to make their steps. Besides of WIFI connection, there is also Bluetooth connection available to connect with the remote system to avoid the disconnection of WIFI. The application will switch to a Bluetooth connection when WLAN is disconnected.

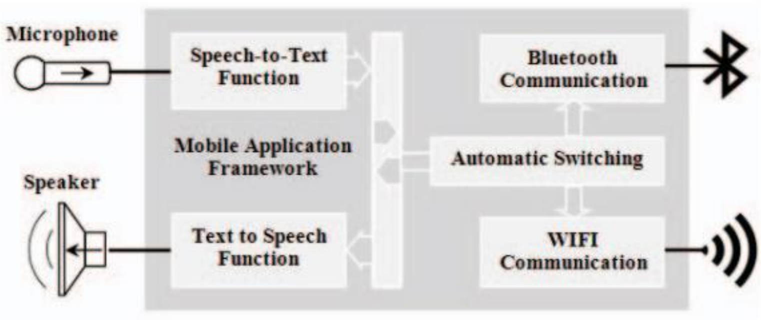


Figure 2.1.3 Mobile Application Functional Architecture [3]

For the remote processing system, there are only two input data sources received by the system that is camera network and the WLAN interface which receive the user voice command in text form. Then, the system will process the two inputs received and generate a text navigation output converted into text and forwarded back to the user's mobile application on the same WIFI interface. There is a built-in image processing algorithm in the system to estimate the position of the user in the respective room and the entire house. This algorithm will only be triggered when there is a motion detected in the house. Then, it will start capturing images from the camera and search for the specific marker that is located at the head of the user. After that, the algorithm starts to allocate and track the location of the user in the house by estimating the coordinates of the marker detected. The coordination can be done by allocating each camera with the Cartesian coordinate concerning the entire floor coordinates. After the location of the user is obtained, the system will analyze and respond according to the user's requirements when receiving a text-converted voice command. Otherwise, the system will only provide the user with obstacles around him by obstacle detection functionality. The real-time image processing technique used by the system is a built-in function in the OpenCV library called “EmguCV”.

2.1.3 Visual SLAM

[4] proposed a system that uses the Visual-SLAM to utilize the RGB image and depth image to perform indoor localization and construct a virtual blind road. The navigation hardware of this system is a combination of fisheye, depth camera, ultrasonic rangefinder, embedded CPU board, and optical see-through (OST) glass. The sparse map will be generated after there are some key positions are tagged in the virtual blind road. Then, the way finding module will depend on the PoI graph, which also means a sparse map to plan the shortest route from the starting position to the destination. The obstacle detection module will guide the user in the walkable depending on the real-time depth image. The ultrasonic rangefinder will help to get the distance of the obstacle. Then the route-following module received the current orientation of the user, possible walkable path, shortest path, and obstacle distance as inputs to generate the guiding information by using the dynamic subgoal-selecting method (Figure 2.1.4).

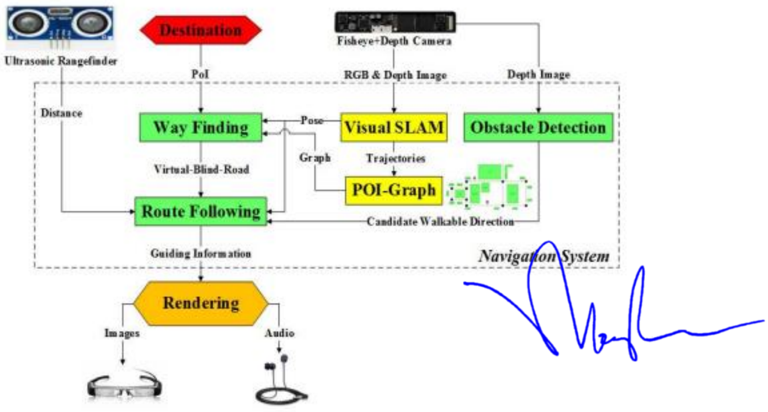


Figure 2.1.4 Architecture of Proposed Navigation System [4]

ORB-SLAM2 was used as it ran three parallel threads. The first thread can perform tracking there is a feature that helps to match the location with the local map of the camera. The second thread controls and runs the local map which is able to insert a new point into the local map and perform local bundle adjustments (BA) to optimize the local map. Next is the loop closing thread, which will compute the optimal global structure and motion solution by launching a fourth thread to perform full BA after a pose graph optimization.

Virtual blind-road and allocation of the use is done by using the ORB-SLAM2. The virtual blind-road is needs to be built before the visually impaired use it. It can be done by a normal-sighted person wearing the glasses and performing threads of the ORB-SLAM2. Figure 2.1.5 shows the virtual blind-road created based on the tracking of the motion by the sighted person. When the system is locating the user, there is only one thread of the ORB-SLAM2 performing the tracking function, and is active from the starting position of the user to the destination.

Figure 2.1.5 Example of the virtual blind road [4]

When the setup of the virtual blind-road is complete, the PoI graph will also be created. The nodes in the graph will represent some regions such as toilets, rooms, and hallway junctions. The connecting edge between the nodes in the PoI graph is equal to the distance in the real world (Figure 2.1.6). The main usage of the PoI graph is to find the nearest path from the user's current position point to the destination by applying the A* algorithm. A* algorithm will search among all possible routes among all the nodes and generate a path with the smallest cost/distance.

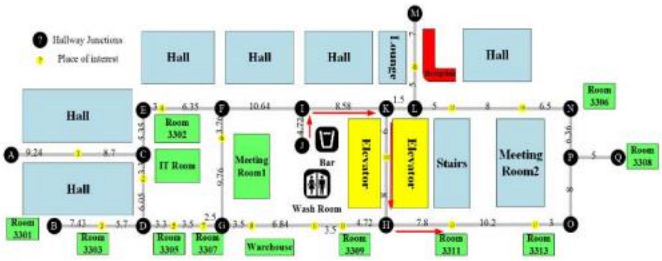


Figure 2.1.6 Example of POI graph [4]

The Route Following Module will receive four inputs such as the shortest path from the way finding module, the current pose from the VSLAM module, the candidate walkable direction from the obstacle detection module, and the range of the obstacle in front of the user from the ultrasonic rangefinder. Then, this module will produce a piece of guiding information for the user to avoid the obstacles on the path by following the shortest path. Figure 2.1.7 shows the process flow of the navigation system. In step 1, to nearest point to the user's current location in the closest route is found. Then for step 2, a subgoal is selected. The subgoal can be a turning point, destination point, or distance from the nearest point. An optimal walkable direction is determined by the subgoal combined with the reliable walking direction produced by the depth image in Step 3. For the last step, the ultrasonic rangefinder is combined to decide the final route direction and output it to the user. These four steps will be looping until the destination is reached (Figure 2.1.6).

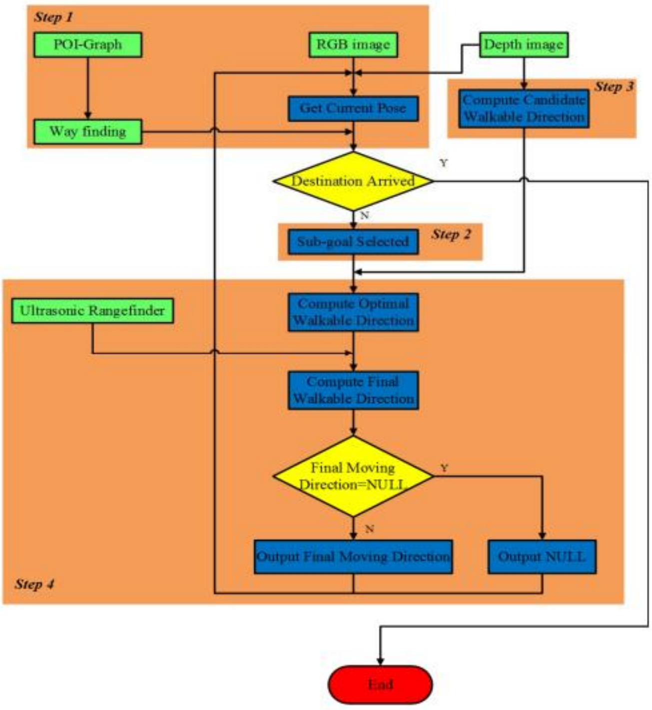


Figure 2.1.7 Workflow of the proposed system [4]

2.1.4 Microsoft Kinect

In the paper [5], the blind navigation system consists of a Kinect camera, earphones, and a laptop. The Kinect camera is used as a depth camera and it provides 30 frames per Second (FPS). Kinect cameras will provide depth information and images from the user's surroundings and send them into the computer laptop program. Then, the program will notify to user by meaningful sound by processing the information data received from the Kinect camera. After that, the sound notification will be output through earphones to the user.

For the preprocessing, it consists of only 3 main steps to filter out noise and improve the detection of the obstacle. To achieve less memory consumption and speed up the processing time, the image received will be converted into an 8-bit unsigned integer from a 16-bit unsigned integer. Then, the image will go through two filters that is median filter and the Gaussian low pass filter. A median filter is used with a neighborhood size of (3,3) because it can remove all possible noise in the image that comes from errors in data transmission. Gaussian low pass filter can take off the Gaussian multiplicative noise. Equations (1) and (2) are used by Gaussian filter. Then, the contrast of the image is enhanced by image histogram equalization.

$$h_g(n_{1,}n_2) = e^{\frac{-(n_1^2+n_2^2)}{2\sigma^2}} \qquad (1)$$

$$h(n_1, n_2) = \frac{h_g(n_1,n_2)}{\sum_{n_1} \sum_{n_2} h_g} \qquad (2)$$

Obstacle Detection Using Window Method

In this method, the image will be divided into 9 windows (sub-arrays) of sizes 213 x 160 pixels (Figure 2.1.8). For obstacle detection, a recursive pattern is used to scan the image for each window and the mean value will be obtained by applying the equation (3).

$$\mu = \frac{1}{N} \sum_{i=1}^{N} A_i \qquad (3)$$

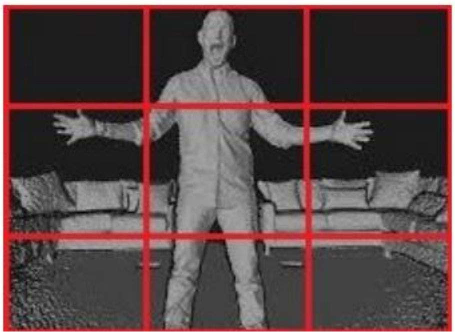


Figure 2.1.8 9 Sub Array Obstacle Detection[5]

Distance calculation

The main contribution of depth image is to determine the range of the obstacle depending on the brightness of the object. When the obstacle gets nearer to the user, the brightness of the obstacle in the depth image will increase. For instance, Figure 2.1.9 (a) illustrates the obstacle is 1.5m away from the user and Figure 2.1.9 (b) illustrates the obstacle is 2m away from the user.

Figure 2.1.9 Depth image for human (a) 1.5 m ; (b) 2 m [5]

For the feedback mechanism, the position of the y-axis determines the frequency of the output sound. the x-axis is used to determine the orientation of the obstacle from the camera (Figure 2.1.10). For instance, an obstacle located at the user's right-hand side will be detected, and depending on the x-axis, it produces the sound for the right earphones only. When the obstacle is close to the user, the volume will become higher to notify the user.

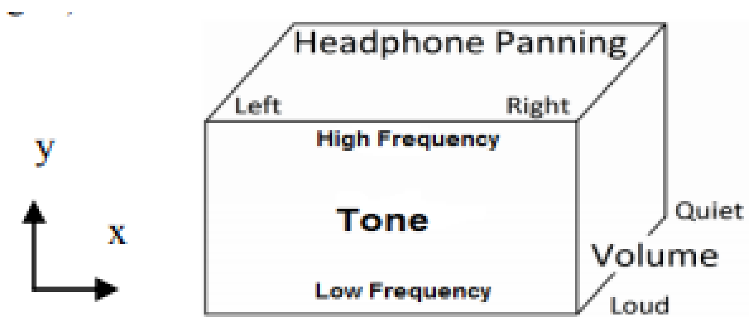


Figure 2.1.10 Transformation of Depth Image into Sound [5]

2.1.5 LIDAR

Echolocation is a natural sensory substitution capability that animals such as bats and dolphins use to detect the location of an object so they can still move around in the darkness to navigate, avoid obstacles, and hunt. This paper [6], proposed an assist spatial sensing (LASS) system to emulate the echolocation by using Light Detection and Ranging (LIDAR) device. The use of the LASS system is to transform the spatial data collected from LIDAR into soundscape information. Computer and stereo headphones are used for processing spatial information and give users a stereo notification about where the obstacle in front of them is.

First, LIDAR will collect the 2D spatial information on a horizontal plane of the user starting from 0 degrees to 180 degrees. The information collected will be saved in (Angle, Distance) format. After that, the LASS system integrates with LIDAR and mounts in front of the user's chest, so the user can scan their surroundings by rotating their body. Then, all the spatial information is transferred and gathered in a computer device for processing by using a cable connection. The computer will generate stereo sound for users with different frequencies that are determined by angular data and distance data. For safety issues, the maximum audio frequency will be capped at 650 Hz to avoid damaging the user's ears because it is the peak frequency that a human can listen to. Then, the sound will be stereo and has various intensities and time delays between the left and right ear to help the user locate the sound source more easily. For instance, in Figure 2.1.11, assuming the obstacle is detected and the system generates sounds from a signal point, the total length of the sound needed to travel to the left and right ear is represented by d1, d2. The formula (4) is used to determine the time difference for sound to reach the user's ears. In this case, the sound will reach the user's right ear followed by the user's left ear since d1 < d2. Besides that, the intensity of sound increases when the distance of the obstacle is decreased. Therefore, the intensity of sound in the right ear will be higher than in the left ear.

$$\Delta t = (d_1 - d_2)/\text{speed of sound} \quad (4)$$

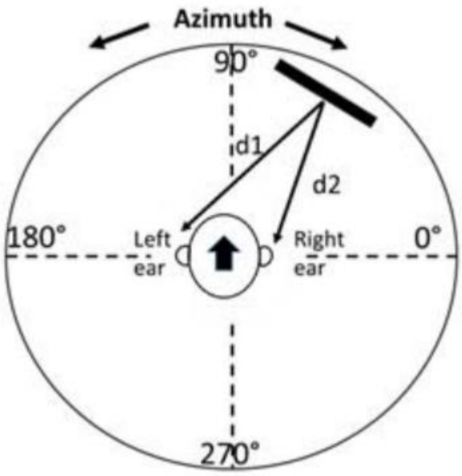


Figure 2.1.11 Localization by sounds. [6]

SFML library is an open-source library that can generate stereo sounds that are encoded with angular information. The angular information received from LIDAR is processed into intensity and time delay through head-related transfer function (HRTF).

Lastly, the LASS system inversely converts distance to audio frequency. When the range between the user and the obstacle decreases, the frequency of the sound would increase. The formula (5) is used to transform the distance into audio frequency.

$$f = aD^2 + bD + c \quad (5)$$

Where f represents the frequency of sound, D represents the distance between the user and the obstacle, and a, b, and c are constant numbers of (650,0), (490,2), and (170,6). Figure 2.1.12 shows what scanned spatial information looks like. Green colour means processes spatial information and red colour means unprocessed spatial data. The user will hear the sound coming from the right side first with a higher pitch followed by the sound coming from the left side. This is because the distance of Obstacle 2 is closer than Obstacle 1.

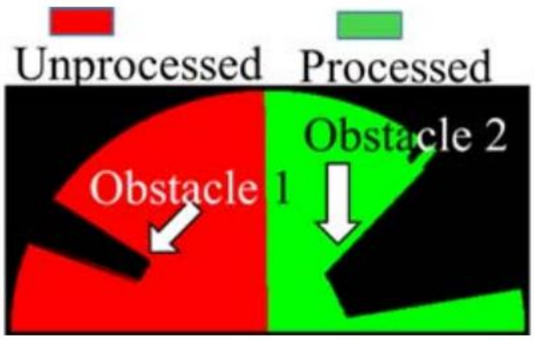


Figure 2.1.12 LASS system scan sample. [6]

2.2 Text and Sign Recognition by Using Smartphones

In this paper [7], text and sign recognition is built by using only smartphone devices for visually impaired people. The smartphone he uses is just an Android phone with a camera and audio system to receive input images and convert the text in images into speech. There are two main processes in the system which are to keep track of texts and signs in every frame of the image generate feedback to the user and use Optical Character Recognition (OCR) to read the result to the user. Figure 2.2.1 visualizes the process flow of the system. As the system flows through the text recognition process, the image processed by ORC will return the raw text from the region that contains text or sign. The raw text will then be sent to another algorithm to separate and correct the raw text input and the result will be converted into text by using text-to-speech algorithms. The process of user feedback also can be divided into three steps. First, all the signs and text will be detected in the input image and then sent to ORC. Then, there will be a guide for the user on how to perform the recognition of signs with the best angle by adjusting the camera. Lastly, commands will be generated and forwarded to the text-to-speech to communicate with the user.

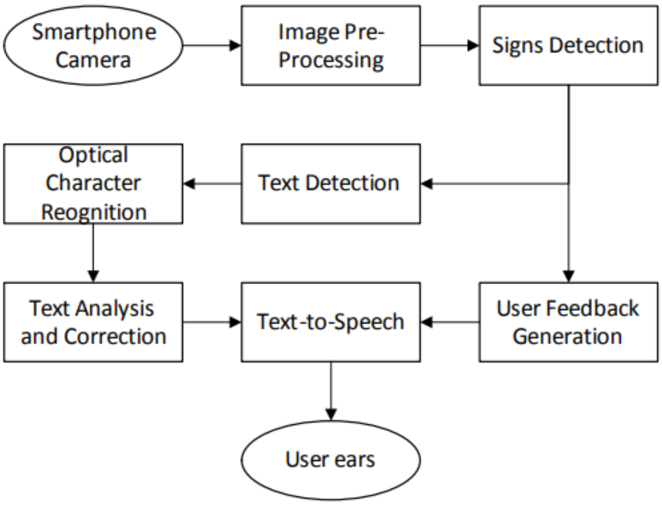


Figure 2.2.1. Block Diagram of the system [7]

The image captured by the smartphone camera will undergo image preprocessing before sign detection. The objective of image preprocessing is to locate and segment all the signs by using colour and shape detection techniques. Then, the ORC algorithm helps to filter out the selections that are irrelevant or have no text sign.

The Tesseract ORC algorithm used by the author is an open-source OCR engine. First, the algorithm uses a line fitting technique to determine the lines, size, and orientation of text in the image. Then, there are two ways to determine the spacing of text which are fixed pitch detection and variable pitch. Fixed pitch uses the method of the pitch is fixed and determined by the average letter separation. Figure 2.2.2 shows a variable pitch that determines the distance between 2 letters, if the distance between 2 letters is more than a threshold, a spacing is added between them. Lastly, OCR needed to recognize each character individually. A neural network in the Tesseract algorithm is used as an adaptive classifier. Additionally, there is an extra step needed to reconstruct or arrange the words and sentences. It can be done by using the information of spacing to group all characters into words. Once completed, they are sent into a Context Grammar algorithm to detect the spelling error of the sentence.

Figure 2.2.2 Variable Pitch technique from [7]

2.3 AI Smart Navigation and Object Detection for Visually Impaired

In this paper [8], the methodology used was a deep learning model, which is YOLO to perform object detection and recognition. Devices such as distance sensors, cameras, and headphones are embedded with a digital signal processor to become a module to run the system Figure 2.3.1 shows the system flow of the system.

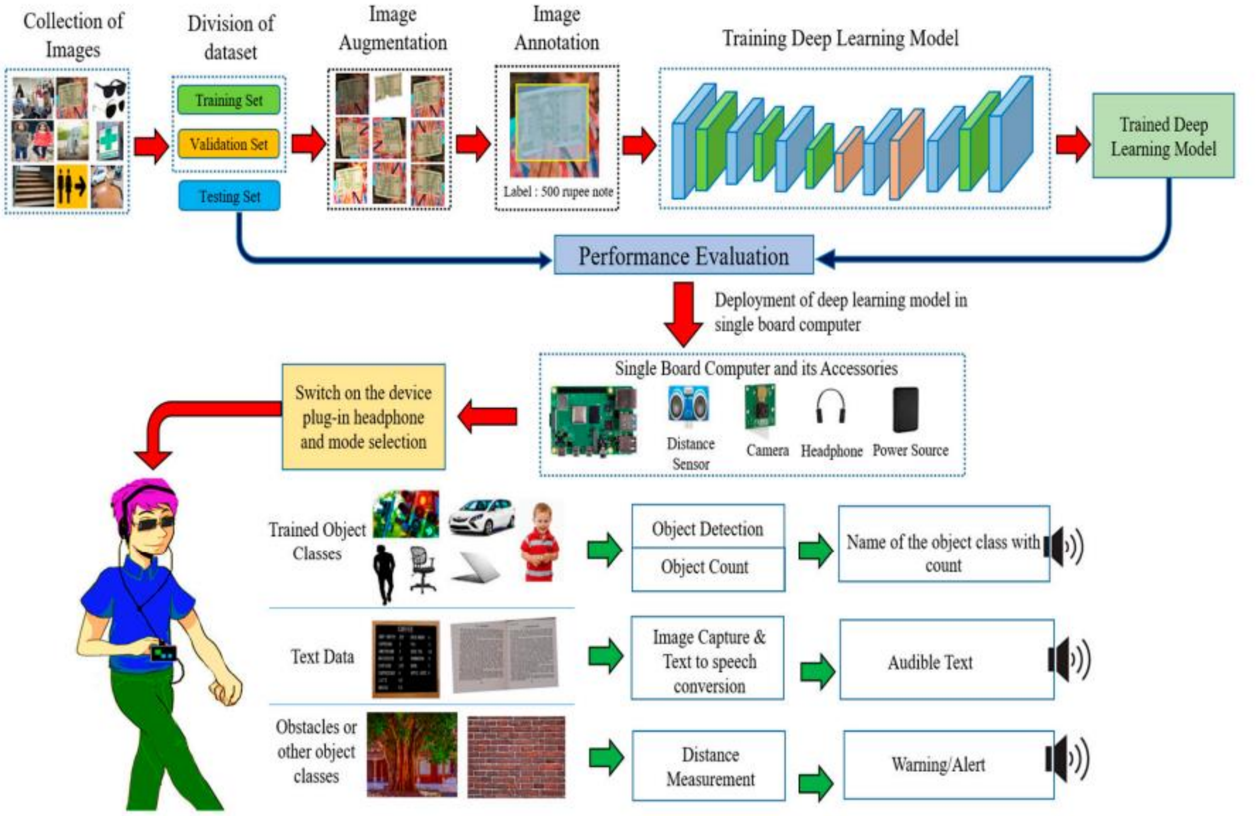


Figure 2.3.1. Block Diagram of System from [8]

Object Detection

First is the preparation dataset for visually impaired people. Images in various sizes and pixels that are generated from different sources and devices with different brightness and capturing angles are selected as a collected dataset for training the model. Then, image augmentation is applied to collected images to enable the system to perform in more stable and higher accuracy conditions. The technique of image augmentation such as skewing, flipping, brightness level, noise level, and rotation of the image was used to reduce the overfitting of the model and enhance the dataset into many folds. After that, the object in the collected images was manually annotated by some labeling tools and a bounding box. The resolution of the image, position, width, and length of the bounding box were saved into an XML file. Then, the YOLO v3 model will be trained by using the annotated dataset and the XML files.

In the process of training the model, the images will be divided into S x S where S is the number of grid cells in each axes. After that, the detection is started by detecting the targets in each unit of grids. Then, the model will detect the presence of any object in the image by outputting a confidence score that represents the likelihood of the object. The confidence score will be 0 if no desired object appears in the image. Else, if an object appears is bounded with a predicted box, the score can be calculated by this equation:

$$CS = P_r(Obj) * IOU^{Predicted}_{Groundtruth}$$

Where CS = Confidence Score, P_r(Obj) = probability of the object, IOU Predicted Groundtruth = Interaction over union (IOU) of predicted and ground truth bounding box. The authors are using Logistic regression to perform prediction of the object score for every bounding box instead of using Softmax.

In the training of the neural network, there are equations to make predictions that are:

$$b_x = \sigma(t_x) + c_x \dots$$

$$b_y = \sigma(t_y) + c_y$$

$$b_w = p_w\, e^{tw}$$

$$b_h = p_h\, e^{th}$$

Where t_x, t_y,t_w,t_h are coordinates that are predicted to be the bounding box, c_x,c_y is cell offset from the top corner of the image, p_h,p_w is the height and width of the bounding box prior.

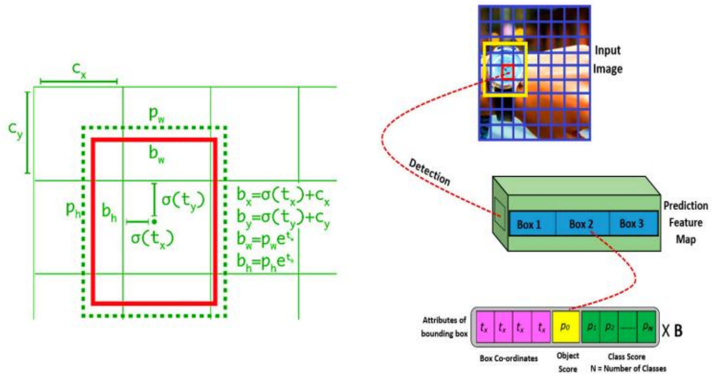


Figure 2.3.2 Bounding box prediction; YOLO-v3 object detection in Bounding Box from [8]

Once this convolutional neural network (CNN) has been trained, a live video is used and each image frame is captured. The Captured image will first be pre-processed and then sent into the trained model. A bounding box will be drawn at the object and the labeling object once there are any objects detected. Once all the object in the image is detected, the text label of the object is converted into speech, then the next frame will be processed.

Image capture and text-to-speech conversation

The module proposed by the authors has multi-functions. The headphone is used to receive the prediction audio prompt. First, there will be no silence in the recording of the object label except for space for all the audio files to ensure the optimization of processing time. Secondly, since the system can detect the same object multiple times, it will accumulate it and output as 5 cars instead of the output car five times. A trained model will be added to count the number of objects with similar categories for each frame of the images. Thirdly, having a large number of object categories takes a long time to notify the information to the user if there are many different kinds of category objects appearing in the same frame. The author solved the problem by limiting the object category to three categories only.

If there is no object detected in the current frame, the system will predict the distance by using ultrasonic sensors. There will be a threshold pre-defined so that when the distance detected is smaller than it, the system will assume there is an obstacle detected. Figure 2.3.3 indicates the combination of object detection and obstacle detection.

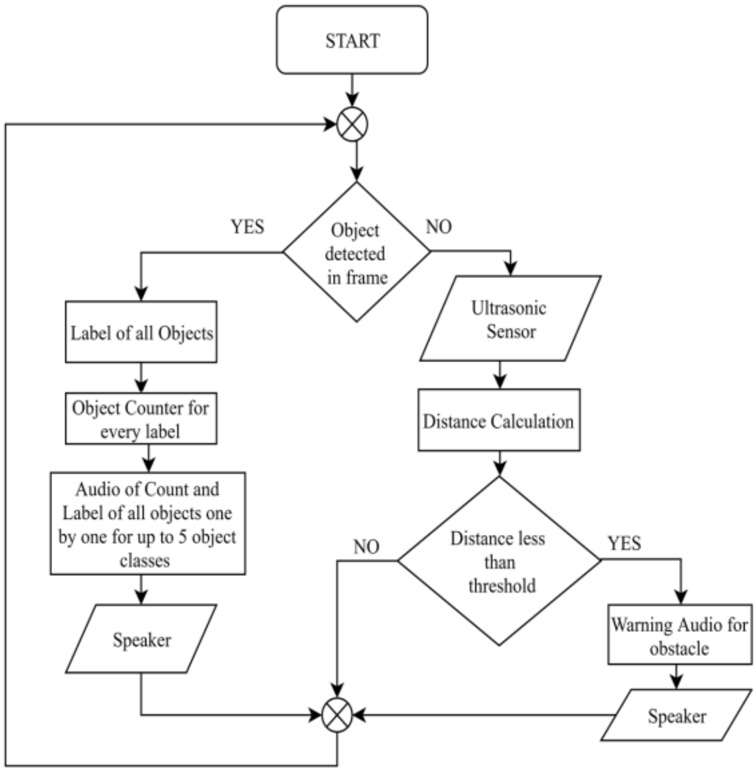


Figure 2.3.3 Flow Chart for Obstacle Object Detection [8]

2.4 Object Detection and Recognition by Using Smart Glass

This author [9] uses smart glass, a smartphone, and a tactile pad to illustrate the surroundings to the user. The architecture used is server-client architecture as a smartphone, smart glass, or tactile pad performs as client side while there is an artificial intelligent server act as the server side to perform object detection and recognition, low-light image enhancement, text-to-speech, salient object detection, and tactile graphic generation. This architecture would save the battery life of the device used because the smart glass only performs image capture and sends it to the smartphone then the smartphone sends it to the artificial intelligence server. Figure 2.4.1 shows the flow of processing images in the artificial intelligence server.

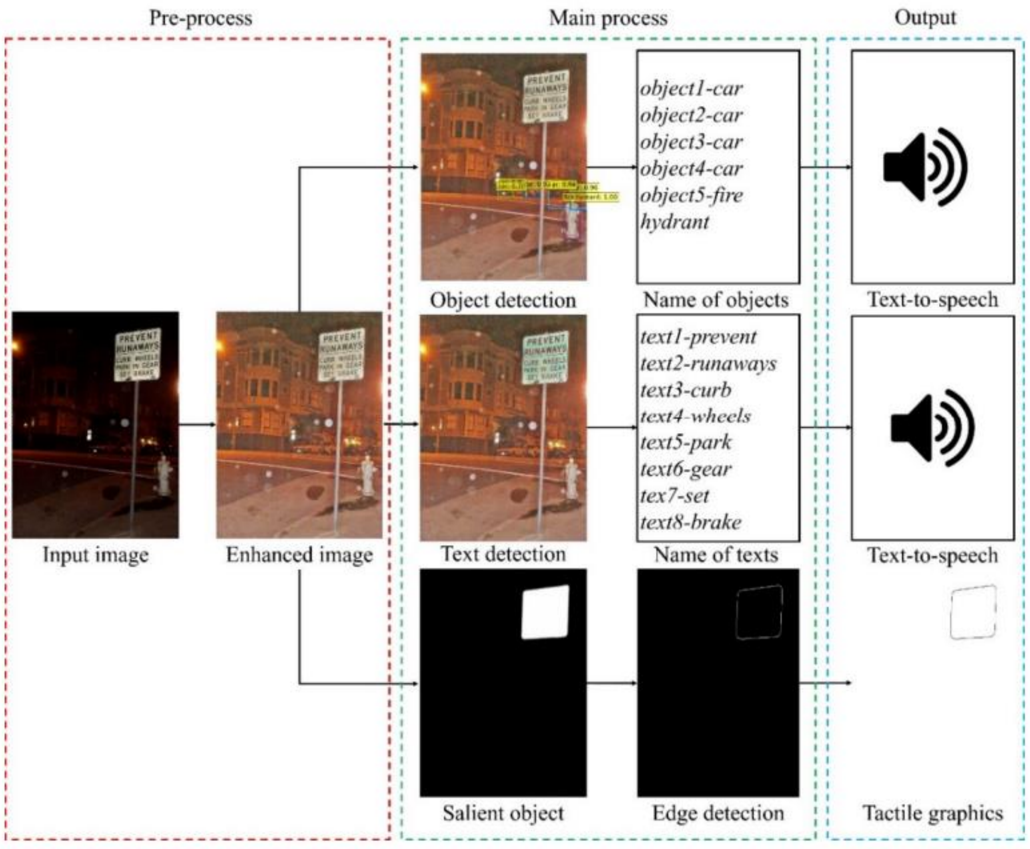


Figure 2.4.1. Data Processing in Artificial Intelligent Server [9]

Once the server receives images, that image will go through the low-light image enhancement model. It helps to remove noise by using a two-branch exposure fusion network that is based on CNN. Figure 2.4.2 shows the image processing flow of the model while the level -1E branch provides a basic enhancement module and the level -2E branch provides a basic enhancement module and preprocessing module.

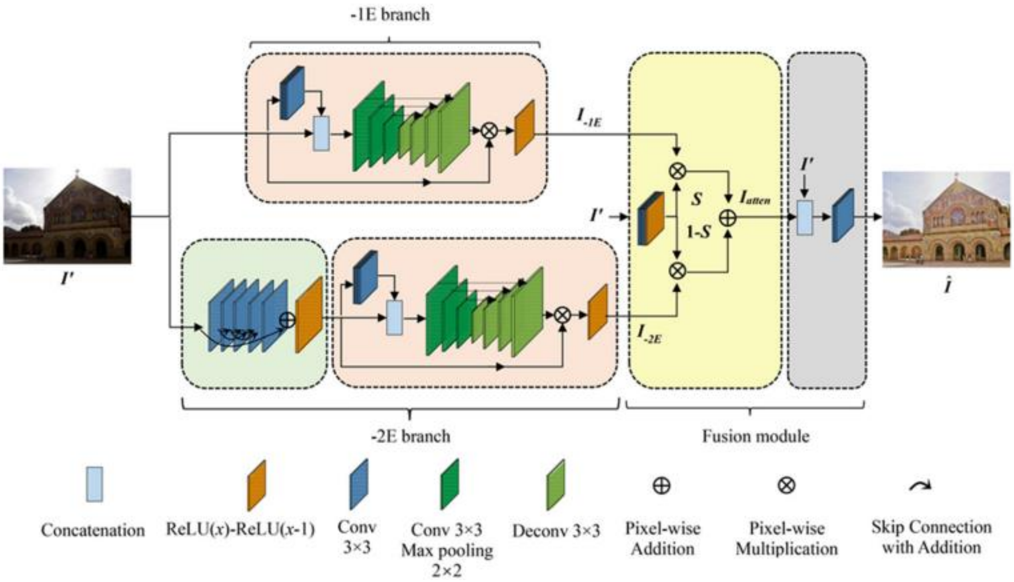


Figure 2.4.2. Low Light Image Enhancement Diagram from [9]

A basic enhancement module is constructed of four convolutional layers of the input image and is utilized to obtain its feature which is subsequently combine with the low-light image input image before being forwarded to the enhancement model. The -2E contains an extra model with a pre-processing module to separate heavily and lightly degraded images that include noise. Multilayer element-wise summations are used in the pre-processing module and five convolutional layers with a filter of 3 x 3 are used and combined with their feature maps to assist in the training process. After that, a fusion module merges the two-branch network to produce the final output.

After that, the object recognition and detection model is deployed by using transformer-based encoder–decoder design. The Detection Transformer (DETR) will predict every object at one time in this model. The network structure of the DETR includes four parts which are the CNN backbone, transformer encoder, transformer decoder, and a simple feedforward network that predicts the output. A lower-resolution activation map made by the CNN backbone is flattened and extended with positional encoding before forwarding to the transformer encoder. Then, the transformer encoder will produce a new feature map by applying a 1 x 1 convolutional filter. After that, the Transformer decoder converts N embedding of size d by using multiheaded self-attention and encoder-decoder attention mechanisms following the standard structure of the transformer. Lastly, a three-layer perception with a ReLU activation function and dimension d, as well as a linear projection layer to produce the prediction.

Besides that, the text-to-speech model was also implemented. The recognition could be a challenge since the image captured by the smart glass might be blurry, colour different, complex background, noise, and lighting problems. The author uses end-to-end scene text recognition that is performed in real-time and a Tesseract OCR engine to achieve a stable and accurate output. The result in dense per-pixel prediction of sentences or text lines can be obtained by a fully convolutional network model for text detection. It can be divided into three stages shown in Figure 2.4.3 that is feature extraction, feature merging, and output layer. The text will finally be forwarded to the TTS model to communicate with the user after being recognized by a trained Tesseract OCR.

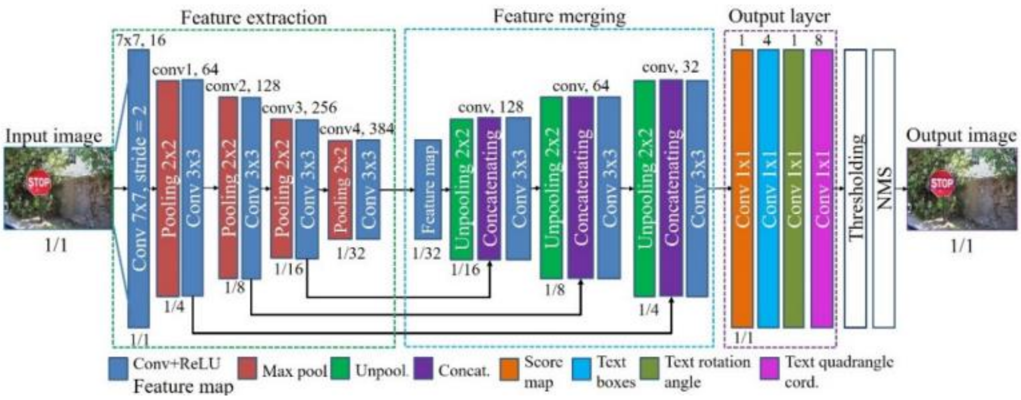


Figure 2.4.3. The network architecture of scene text detection diagram [9]

2.5 Indoor Navigation and Tours for Blind by Using Mobile Application

[10] is an Android phone application that is implemented to guide people with weak eyesight or blindness in the Museum Blind MusuemTourer application. The use of earphones is recommended due to the application will keep guiding the user with voice navigation instructions. The user can give commands to the application through voice commands or widgets on the smartphone screen. The user would receive a voice instruction about how to interact with the application once the application starts. Figure 2.5.1 shows the example of voice guidance once the application starts. After that, Table 2.5.1 is an example of navigation instructions from Blind MusuemTourer.

"Welcome to the museum self-guided tour application!

- *For your convenience, you may interrupt the guided tour anytime by double tapping at the upper screen to move to the restroom, the cafeteria, or the exit, to talk to the help desk or make a phone call.*
- *Anytime you wish to go back to the previous menu, double tap at the bottom screen.*
- *To select an option double-tap the respective left, middle, or right section at the centre screen.*
- *To hear a selection, please single-tap at the respective screen section."*

Figure 2.5.1 Guidance to the user when starting the application [10].

Sequence No.	Navigation Instructions
1	(User has entered the building and is approaching the Reception Desk). "Stop. Reception reached." (System waits until user has stopped). (Possible adjustment follows, e.g., "Move 1 step/small step in front/behind.") "Position at 3 o'clock." "Exhibit 1 next."
2	(User moves towards next exhibit). "Mind the short stairs in front." "Exhibit 1 in 2 steps following the short stairs." "Stop. Exhibit 1 reached." (System waits until user has stopped). (Possible adjustment follows, e.g., "Move 1 step/small step in front/behind.") "Position at 10 o'clock." (Narration). "Exhibit 2 next in 13 steps following the right turn."
3	(User moves towards next exhibit). "Stop. Exhibit 2 reached." (System waits until user has stopped). (Possible adjustment follows, e.g., "Move 1 step/small step in front/behind.") "Position at 10 o'clock." (Narration). "Exhibit 3 next in 15 steps."
4	(User moves towards next exhibit). "Stop. Exhibit 3 reached." (System waits until user has stopped). (Possible adjustment follows, e.g., "Move 1 step/small step in front/behind.") "Position at 2 o'clock." (Narration). "Exhibit 4 next in 7 steps."
5	(User is interrupting the tour asking wayfinding instructions to the rest room).
6	"Turn left on the sign." (User moves along the tactile path) "Turn right on the sign." (After exhibit 14 is reached and direction has changed through sensing the warning sign.) "Rest room next in 9 steps." "Stop. Rest room reached." (System waits until user has stopped). (Possible adjustment follows, e.g., "Move 1 step/small step in front/behind.") "Position at 3 o'clock."

Table 2.5.1 Blind MuseumTourer navigation instruction example [10].

Tactile walking surface indicator (TWSI) is a physical way to help blind or visually impaired travel indoors and outdoors that is detectable underfoot when walking or by using a cane. Author [10] uses an indoor user positioning technique that uses beacons as proximity sensors and puts them on the exhibit and other guided spots in the museum with TWSI mounted in the museum. Figure 2.5.2 represents the floor plan with the TWSI and proximity sensor installed in the museum room. The length of the route is calculated in the method with the x and y dimensions of the room. When the user approaches the exhibit in the museum, the proximity sensors of the exhibit will inform the position of the user to the application and perform a vocal presentation. To interact with the sensor that is located on the exhibits, a small amount of the data will be transmitted toward the low-cost Bluetooth beacon at the exhibits from the application.

The application was able to keep tracking the current position of the user while performing an autonomous guidance route by receiving the beacon signals.

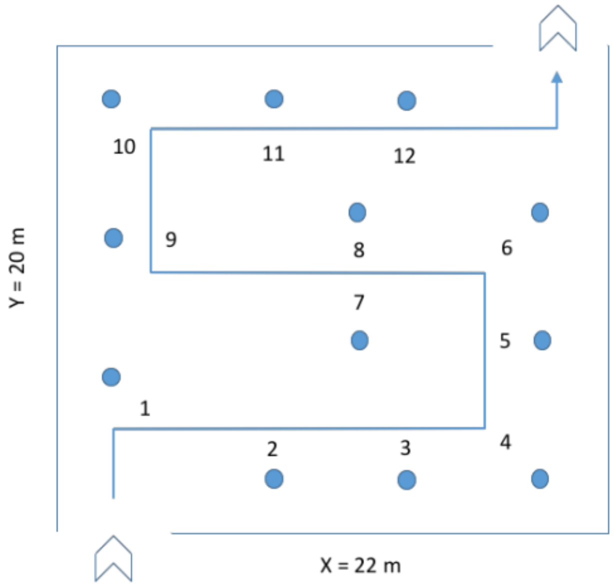


Figure 2.5.2 Placement of Surface-Mounted tactile and beacon signals [10].

Inertial Dead Reckoning (IDR) Calculation

Besides only using beacons and TWSI, an inertial dead reckoning calculation will also be performed continuously while locating the user position. One example of IDR is visually impaired counting the steps of the user and distance calculations such as walking or turning recognition. Therefore, the accelerometer in the smartphone plays an important role in continuously sampling user x, y, and z coordinates while traveling in the museum. The system will calculate and monitor the pace of the user from one exhibit to another and the average stride length to perform a precise navigation instruction to the user. Figure 2.5.3 shows the sample of the calculated accelerometer value. The accelerometer can also detect the motion of the user such as stopping or accelerating accurately in real time. With the data recorded about the range between exhibits, the application can predict and notify users about the remaining number of steps to reach the next exhibit. The application is also able to detect the turning motion of the user with the mechanism of combining the usage of the rotation vector and accelerometer. This mechanism reports azimuth, pitch, and roll value while azimuth and pitch are used to detect user turning motion. After that, the author integrated the

IDR mechanism with a magnetometer sensor in the smartphone to detect the direction of the user going through the user device that depends on the Earth’s magnetic north.

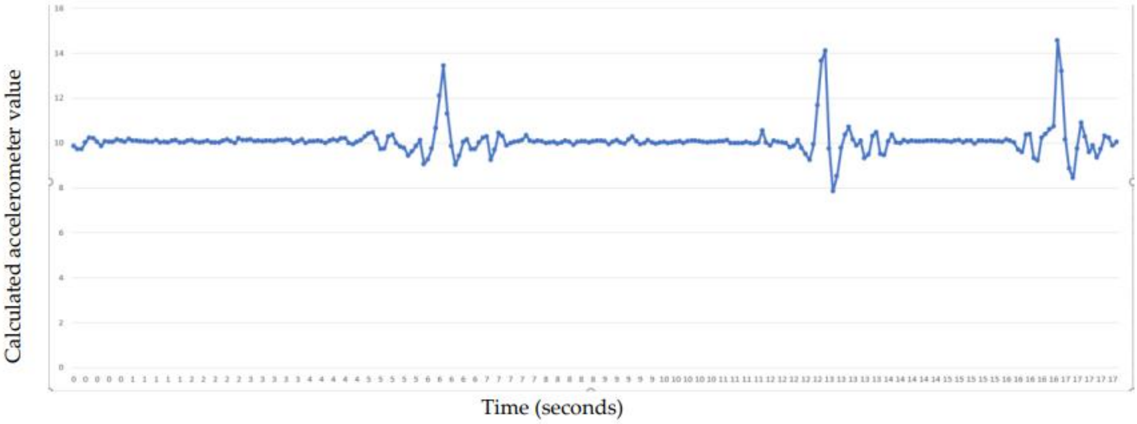


Figure 2.5.3 Calculated sum squared values of accelerometer sensor coordinates over time during three steps with short rests in between [10].

Indoor space map and positioning

A two-dimensional tile map is used by the application to represent the indoor spacing and museum hall. The area of each tile is fixed to 1.5 x 1.5 m^2 with its own x and y coordinate. The location of the user will represented by a red colour tile with its Boolean flag that is true according to the estimation of combined dead reckoning calculations and beacon proximity. Moving toward another tile turns the previous tile's Boolean flag false and white colour. The Boolean flag of the tile will be set to -1 when there is no assistive tactile path in the area. Figure 2.5.4 illustrates the map created and there are three beacons located along the tactile path are detect the current position of the user. Gray colour tile represents the area with no assistive tactile, yellowish circular shows the potential range of user positions that can be detected by beacons. When there is a case of more than one beacon detected by the user, the nearest beacon is responsible for estimating the current user position.

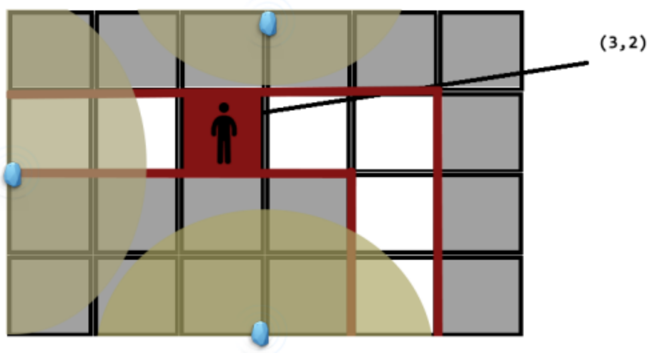


Figure 2.5.4 Beacon detecting current indoor BVI user position [10].

2.6 Limitations of Previous Studies

In this section, the limitations of each reviewed system will be identified. The limitations will be summarized and arranged in tabular form.

Previous Studies	**Limitation**
Stereo Camera	• The system works well in the outdoor environment but not in the indoor environment. • A white cane is still needed since there is no accurate distance and object details provided to the user. • No proper guidance to navigate the user to their destination
IP Camera Network	• Expensive in the installation of multiple IP cameras.
Visual SLAM	• There is a chance to misguide the user because ultrasonic sensor readings depend on the change in temperature and humidity. • Assistance from others is still needed in a new environment for building Virtual Blind Road. • High cost in hardware configuration.
Microsoft Kinect	• Kinect is sensitive to sunlight brightness. • Cost inefficiency. • User just barely knows the location of the obstacle by guessing the sound frequency.
LIDAR	• It needed 13 seconds to complete a full scan. • User just barely knows the location of the obstacle by guessing the sound frequency. • Expensive in hardware.

Text and Sign Recognition by Using Smartphones	• The system cannot detect the symbol in the sign such as the arrow sign. • The worst angle of scanning would cause the sign incorrectly detected.
AI Smart Navigation and Object Detection for Visually Impaired	• Need a huge number of augmented image datasets to train the model. • Cost inefficiency.
Object Detection and Recognition by Using Smart Glass	• Wrong detection of the object may occur when many objects are appearing at the same time. • Text recognition has a certain error when the character is small, orientation and character is being blocked. • Expensive in hardware.
Indoor Navigation and Tours for Blind by Using Mobile Application	• The creation of a map is time-consuming for collecting map measurements and installation of beacons. • Changes needed to apply to the created map when there are changes in the environment. • Cost inefficiency.

Table 2.6.1 Summary of Limitations of the Reviewed System

Most of the reviewed systems lack the ability to locate the user's current position. It is important for the visually impaired to know their current location for a better navigation experience. In case the system is out of battery or malfunctions, the visually impaired can still understand their current location clearly and will not be confused and panic.

Moreover, the hardware requirements from most of the reviewed systems were expensive. Not every visually impaired was able to purchase smart glass or LIDAR devices for the system. From [12], the World Health Organization (WHO) stated that

90% of the visually impaired were from low to middle-income countries. Therefore, the visually impaired will try to avoid the indoor navigation system with the high cost needed.

2.7 Proposed Solutions

This project aims to propose a solution to navigate the visually impaired in an indoor environment that allows them to travel indoors independently. The core of this system is to detect the door label by implementing computer vision techniques. After detecting the door label, the user's current location will be generated based on the door label and the current location will be conveyed to the user by voice message. After that, there is needed a path planning system to lead the user to their destination. The path planning system is built by the graph approach which will assign each room as a node and connect all of them by edges. The weight of the edges between the nodes is represented as the distance between one room to another. When the system receives user instructions, the system can provide speech navigation instructions to navigate the user with the shortest path.

CHAPTER 3

System Methodology / Approach

3.1 System Architecture

Figure 3.1.1 shows the overall system architecture diagram of the implemented system. There are a total of 7 modules to help the system to get the job done. Those modules are the input module (Input Tools), image processing module (Red Dot Detection), word extractor module (Door Label Reader), guiding module (Indoor Navigation), converter module (Text and Speech Converter), and output module (Output Tool). First, the input module will capture real-time image data with the help of the webcam, and it can also receive and record the user voice input by using the built-in microphone of the laptop. Each captured frame will be sent into the image processing module to find and segment out the door label in the captured frame. Once the door label has been segmented out from the frame image, it will be sent into the word extractor module which will extract the words that are written on the door label. Once the system gets the door label name, it will send the name into the guiding module to check the existence of the door label name in the building and update the user's current location in a pre-built map. If there is any user voice command, the guiding module will generate a guiding instruction based on the user's destination and return to the system. Else the guiding module will only return the door label name to the system. After that, the output from the guiding module will be sent into the converter module to convert the text into an mp3 file voice message and return it to the system. Lastly, the voice message will be output to the user by using an earphone.

Input Module (Input Tools)

There will be two hardware components used in this module which are a webcam and a microphone. The webcam will be mounted on a cap to capture the image data and the microphone will be receiving user voice commands after triggering the voice input key (press any key). The webcam will return the image frame to the system and the microphone will return audio data to the system.

Image Processing Module (Red Dot Detection)

The system will send the captured frame into this module. The module will start to locate the red colour dot in the image. First, the module will look for the red colour region in the frame image. If there is any red colour region, the next step is to detect the shape of the red colour region. Once the shape of the red colour region is detected as similar to the shape of a circle, the module will record the red circle region to perform the image segmentation and generate a ring bell sound to the system to inform the user to slow down for a clearer image to be captured by the webcam. The image segmentation will cut out most of the background information which is useless information to enhance the performance and accuracy of the process later. Then, the segmented frame image will be sent into the word extractor module for further processing.

Word Extractor Module (Door Label Reader)

This module will recognize the words in the segmented image and output as a string. The segmented image will locate the region that possibly contains the alphanumeric. Once all the regions are located, the Tesseract OCR will further segment every single region to analyze each alphanumeric individually. Then, the Tesseract will recognize the characters one by one from the segmented images. The predicted characters will then combine according to the sequence that same as in the image and return to the system.

Guiding Module (Indoor Navigation)

This module contains a building floor plan map in a weighted graph format. The module will first check the existence of the door label name in the map once receives the name from the system. Then, the module will update the user's current location on the map and return the user's current location. If there is any user command to guide them to their destination, this module will generate the shortest path from the user's current location to their destination. Once the path is all set, this module will generate some simple instructions such as “walk forward”, “turn left”, “turn right”, and “turn backward” and return to the system. The reason to keep the instruction simple is to not confuse the user while navigating the user's walking direction.

Converter Module (Text and Speech Converter)

This module will differentiate the format of the input from the system. If the input is words, it will convert it to a mp3 file and return to the system with the help of Google Text to Speech. If the input is in audio file format, Google Recognizer will be used to convert the voice audio file into string format and return it to the system.

Output Module (Output Tool)

This module will generate sound based on the input from the system through the hardware component of headphones.

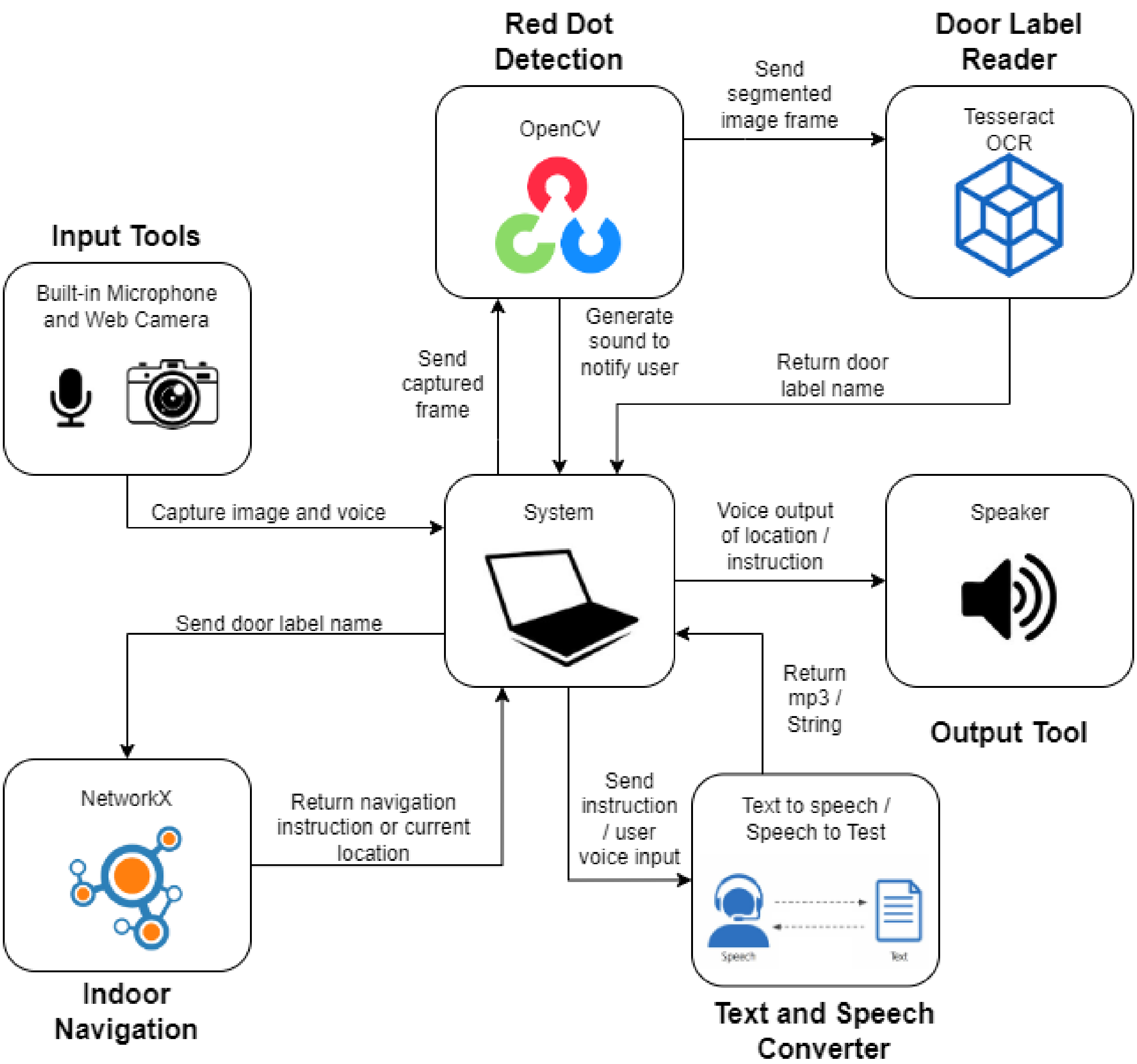


Figure 3.1.1 System Architecture Diagram

3.2 System Flow

Figure 3.2.1 shows how the system flows to process the user input. When the system starts, the system will ask for the user voice input command. This can be the user's destination or nothing. Then, the webcam that attaches to the user will start and capture the image frame. Once the image frame is captured, the system will perform the red colour dot detection. If no red colour dot is detected and the user continues using the system, it will loop back and capture the next frame to find the red colour dot. If a red colour dot appears in the frame image, the system will play a ring bell sound to inform the user that there is a door label and slow down their step for more accurate detection. Then, the system will recognize the door label name. If there is a user command, the system will calculate the shortest path to the user destination and generate instructions. Otherwise, the system will just voice out the door label name to update the user their current location. If the user presses to exit, the system will be terminated.

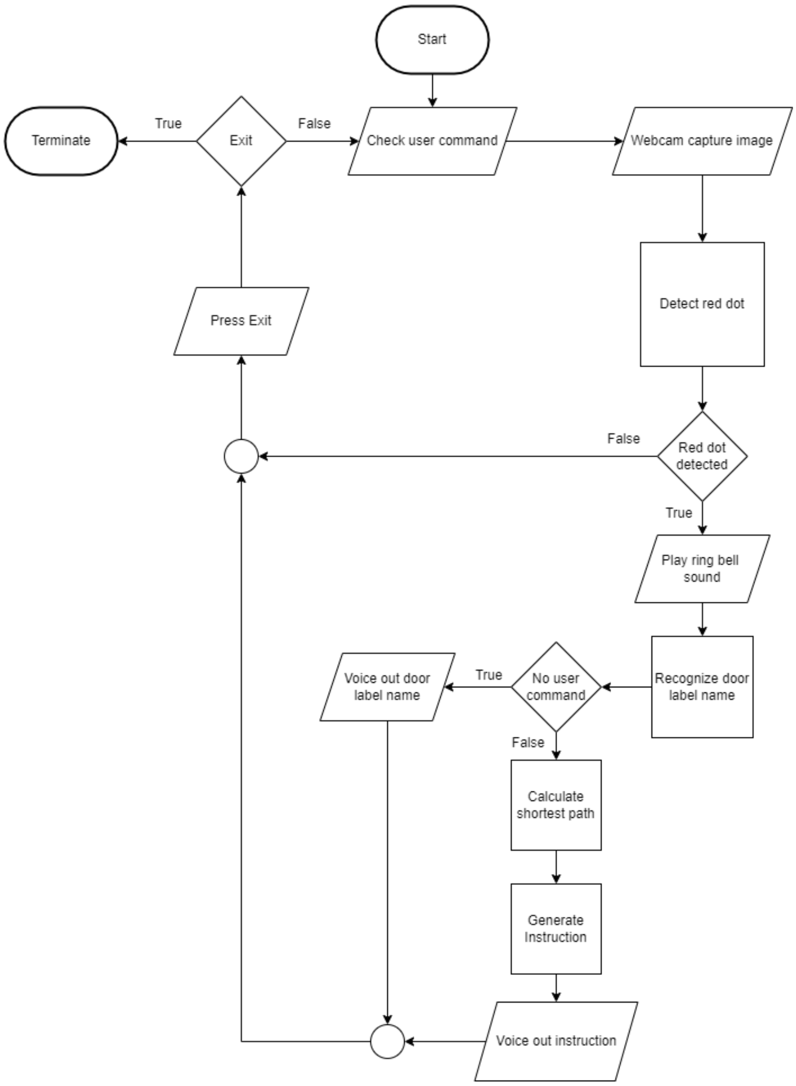


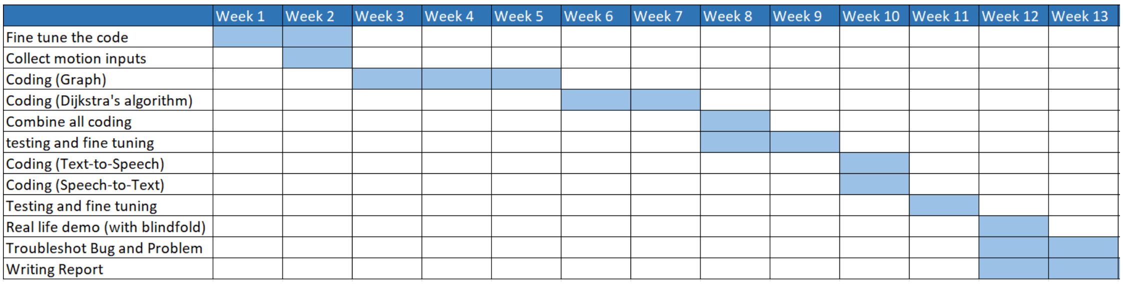
Figure 3.2.1 System Flow Diagram

3.3 Timeline

The project planning and implementation timeline of my FYP1 and FYP2 are shown in Table 3.3.1 and Table 3.3.2.

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
Study about Image processing	■	■				
Study about similar system		■				
Coding (Color Detection)			■			
Coding (Circle Detection)				■		
Coding (Tesseract OCR)					■	
Writing Report						■

Figure 3.3.1 Timeline in FYP1



Fidure 3.3.2 Timeline in FYP2

CHAPTER 4

System Design

4.1 Door Label Detection

To let the system locate and differentiate the door label from the background more accurately, a red colour dot was placed beside the door label to indicate that there was a door label. This means the accuracy of the red colour dot detection will be crucial in this system. The reason for using a red colour dot as the coordinate to locate the door label is that red colour is not a common colour that will appear on the wall and door. Therefore, red colour can be quickly differentiated from another colour in the image frame. The door label will be segmented out based on the location of the red circle dot.

After segmenting out the door label based on the position of the red colour dot, the segmented image will be sent into the Tesseract OCR to recognize the words on the door label. Inside the Tesseract OCR, the image will be further segmented into pieces based on the region that has a high possibility is a character. After that, the Tesseract OCR will recognize the characters one by one based on the segmented image. The block diagram of the door label detection is shown in Figure 4.1.1.

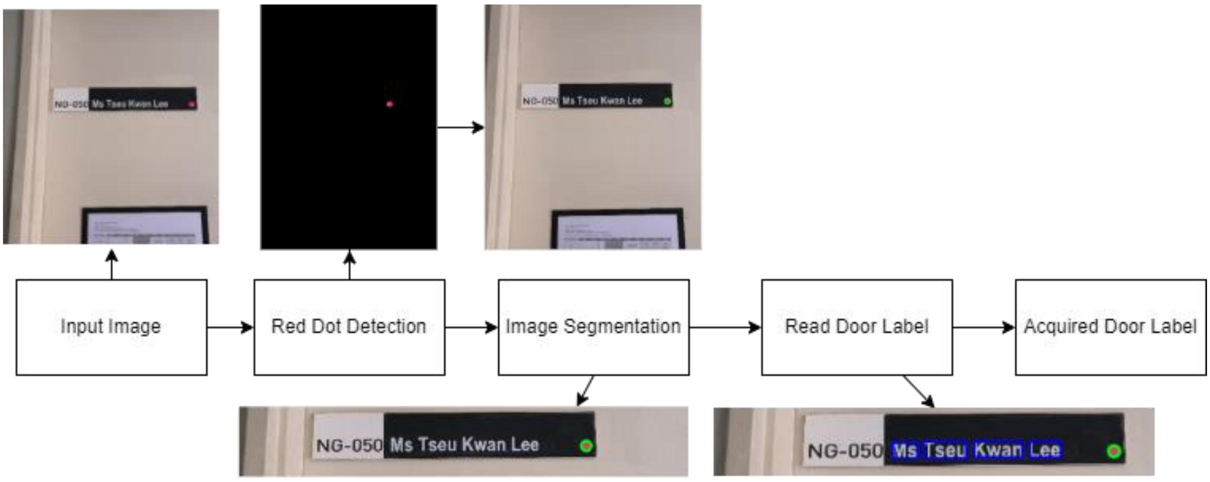


Figure 4.1.1 Block diagram of reading door label

4.1.1 Red Dot Detection

Figure 4.1.2 shows the block diagram of the red dot detection. When receive the image frame from the webcam, we convert it into HSV colour image. Then we generate the mask image by using a defined HSV colour range. The output from these two steps is we will obtain the image frame that contains only the red colour region. Then, Houghcircle in OpenCV is used to do the circle detection on the red colour region only to reduce the computational force of the system. Once the circle is detected, it will generate the circle mask which masks the red colour dot in the image frame. Then we can locate the coordinate of the red colour dot in the image frame based on the circle mask.

Once the system successfully detects a red colour dot, the system will start the segmentation process automatically. First, the position of the red colour dot will be used as the coordinate to perform image segmentation. Then based on the coordinates we get, we segment the image by 80 pixels above and below. This range can successfully segment out a complete door label without missing any part of the door label. We only segment the image frame vertically based on the center of the red colour dot. The reason we did not perform rectangle detection to further segment out the perfect door label image is to prevent the case webcam only capturing the image frame of only half of the door label. This will cause the rectangle detection not to function and miss the chance of segmenting the door label for OCR. Figure 4.1.3 illustrates the process of the red dot detection of the system.

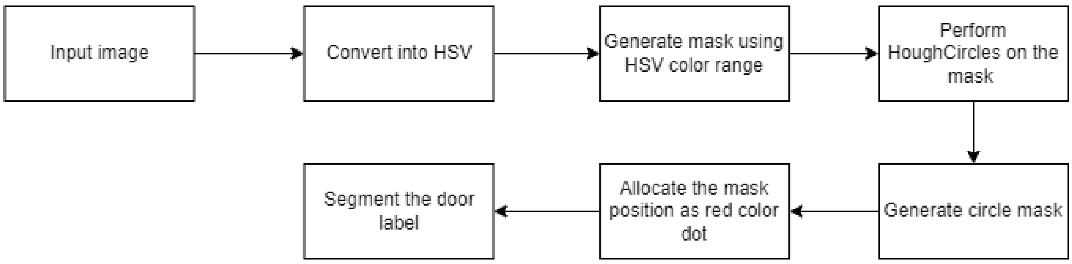


Figure 4.1.2 Block diagram of red colour dot detection

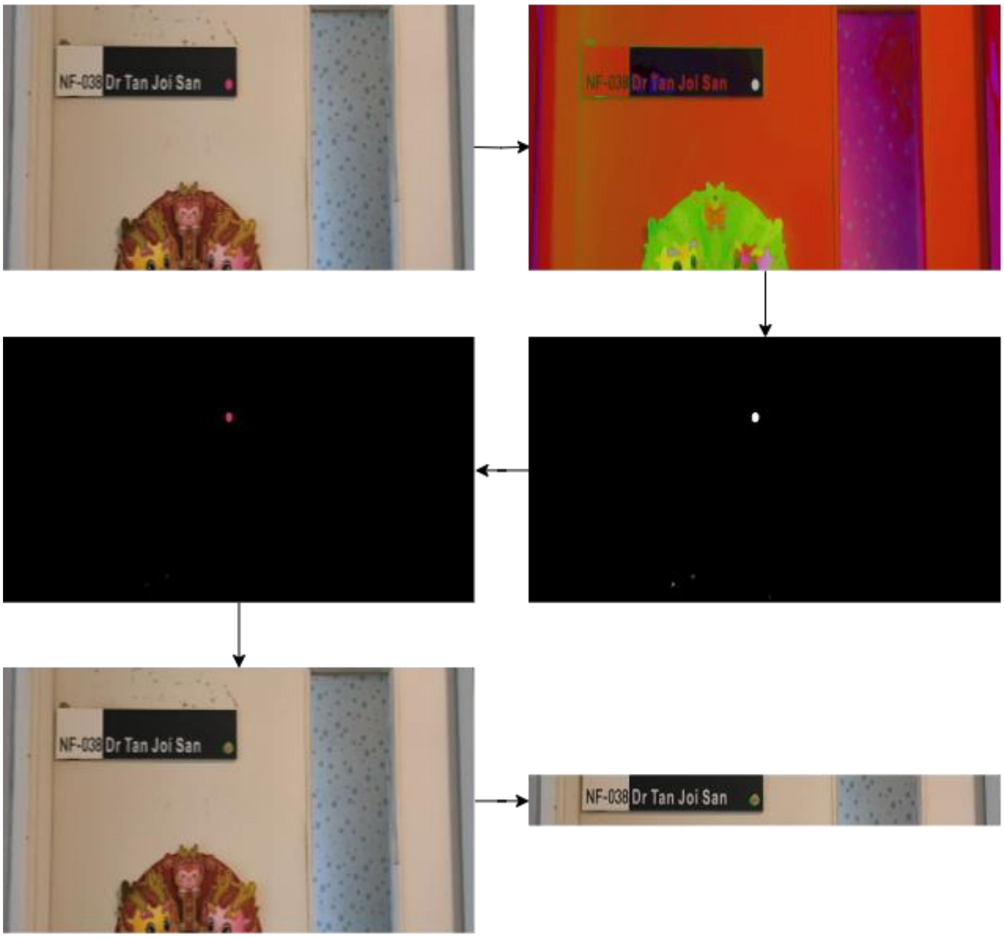


Figure 4.1.3 Visualization of the red colour dot detection process

4.1.2 OCR Reader

Figure 4.1.4 shows the block diagram of reading a door label. After the image is segmented, we first turn the image into a grayscale image and then to a binary image. The reason for turning the image into a binary image is to eliminate the noise that might appear around the door label. Then, the image will be sent into the Tesseract OCR to perform character recognition. After we get the output from the Tesseract OCR, FuzzyWuzzy will be used to compare the output with a list that contains all the door label names in the building to get the full name of the door label. This step is to ensure that the full name of the door label can be delivered to the user. Figure 4.1.5 illustrates the process of reading the door label.

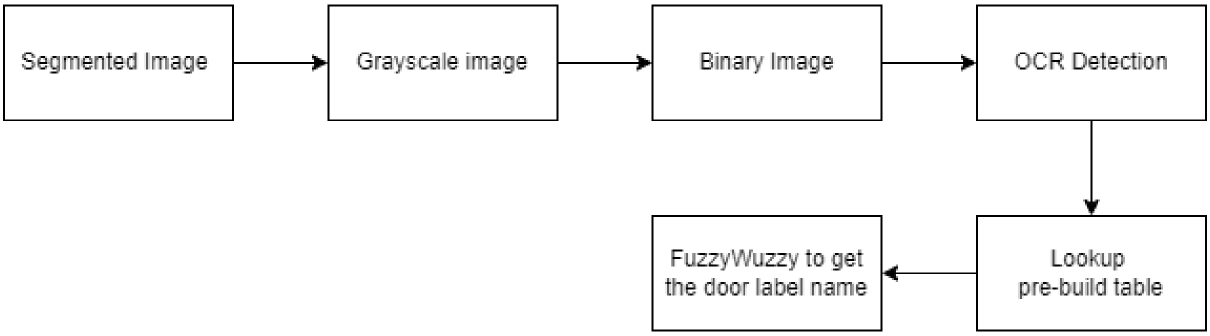


Figure 4.1.4 Block diagram of reading door label

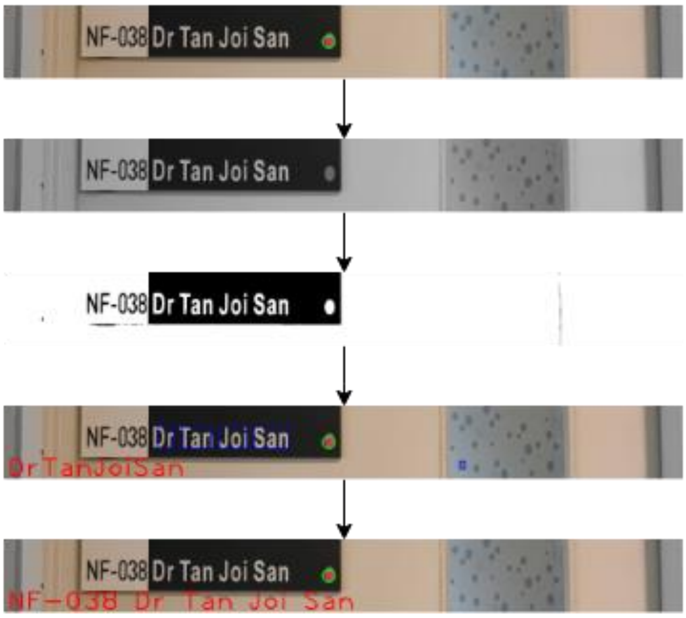


Figure 4.1.5 Visualization of reading door label process

4.2 Indoor Navigation

Figure 4.2.1 is the block diagram of the indoor navigation. Before we can guide the user from point A to point B, a map is needed for the system to keep tracing and updating the user's current position so that it can generate guiding instructions to give the user an overall direction to reach their destination. Therefore, a pre-build graph that is similar to the building floor plan is a necessary item in this system. When the system receives the user's destination, the system will record it and send it to the Google Web Speech to do the conversion and send it back to the system as a string. Once the system receives the voice string, FuzzyWuzzy will be used to get the correct door label name and then search for the node position of the user destination. The system will then generate the instruction to the user after successfully locating the user's current location in the graph map. Every time a new instruction is generated, the new instruction will be voiced out to the user to inform the user of their walking next walking direction.

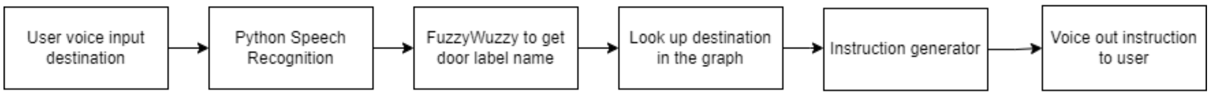


Figure 4.2.1 Block diagram of indoor navigation

4.2.1 Graph Implementation

For the graph implementation, figure 4.2.2 shows the implemented graph map in UTAR FICT. NetworkX is used to draw a graph by using graphs in the data structure method. In the graph, there will be nodes and edges. Each node is assigned as one of the rooms in the building. The edge between two nodes represents the approximate range of the path. Once the user detects a door label, it will be updated to the graph and a pointer will be pointing to the node as represented as the user's current location. Then, if the user gives a command to go to the destination of Lecture Room 4 from his current location, another pointer will be pointed toward the destination node.

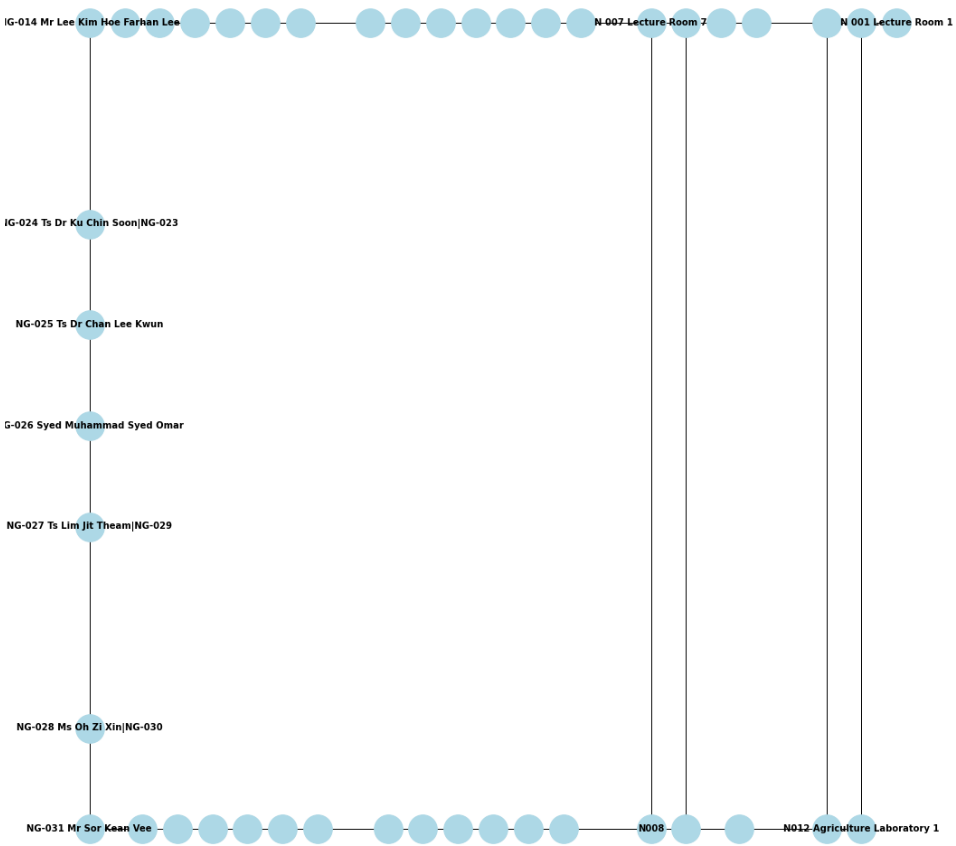
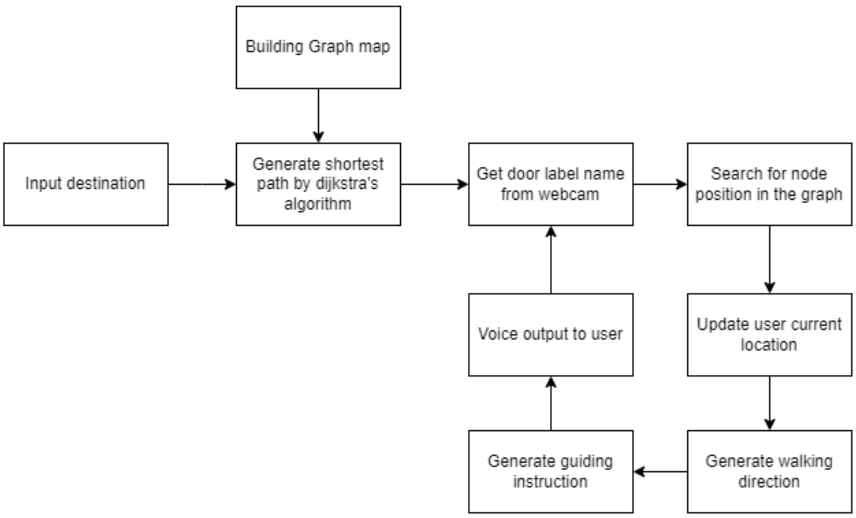


Figure 4.2.2 Graph map of the FICT building

4.2.2 Instruction Generator

Figure 4.2.3 is the block diagram of the instruction generator. After we had our pre-built graph map ready, we were ready to generate the instructions to guide the user from point A to point B. Once the user inputs the destination, the system will quickly look up the destination node in the map and plan the trip for the user with the nearest path to reach their destination. The path will be generated with the help of Dijkstra's algorithm. Then, every time the webcam detects the door label and is successfully recognized, the recognized door label will be used to search for the user's current location and update the system. Then it will generate walking directions based on the shortest path then provide the user a correct instructions to continue their journey. This looping will stop until the user has successfully reached their destination.



4.2.3 Block diagram of instruction generator

Speech to Text

Figure 4.2.4 shows the block diagram of converting speech to text. To enable the user to interact with the system by using their voice, a speech-to-text converter is needed so that the system can understand the user's needs. The moment that the user presses the navigation button, the system will ask for the user's voice input by voicing out "Hi, where do you want to go?". During the system voices out the question, the system will immediately initialize the use of the microphone to receive the user input. Then, the system will clear the ambient noise from the surroundings to focus on what the user said. This can be done with the help of the Python speech recognition library. Then, the audio will keep recording until there is no more user voice received. After that, the received voice message will be sent to the Google speech-to-text server to do the translation and conversion. After the conversion is complete, the Google server will return the string data type to the system.

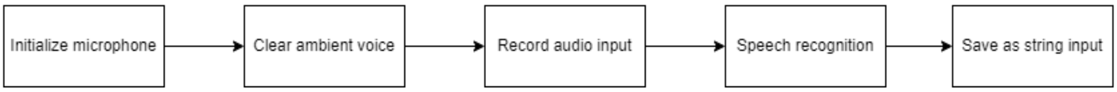


Figure 4.2.4 Block diagram of speech to text

Text to Speech

Figure 4.2.5 is the block diagram of converting text to speech. To enable the system to talk to the user, a text-to-speech converter is necessary in this system to convert the string value into a mp3 file. Once the correct door label name is obtained or the guiding instruction is generated, the system will send the string value into the Google Text to Speech for recognition purposes. Then, Google Text to Speech will record the string input in a mp3 file and return it to the system. Access to the speaker is needed to play the mp3 file to the user. Therefore, Pyaudio will then initialize and load the mp3 file into the mixer and play the mp3 file to the user.

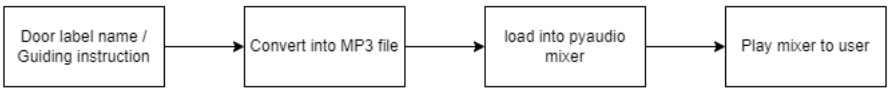


Figure 4.2.5 Block diagram of text to speech

4.3 System Component Interaction Operation

There are only 4 hardware components used in this project which are a web camera, earphones, a built-in microphone, and a computer. Web cameras will act as human eyes to help the user to “see” his/her surroundings. The web camera and earphones will be connected to the computer for data transmission. The web camera will transmit the visual data of the user's surroundings to the camera into the computer. The microphone will capture the user's voice and send it to the computer. The computer will start to process all the input from the components and generate voice output for the user by using an earphone.

The devices used in this project are just a simulation of the higher-grade equipment. For example, the computer can change into some mini-computers such as Raspberry Pi and Arduino. These mini-computers are easier to carry for the user compared to carrying a whole laptop while walking around inside the building.

CHAPTER 5

System Implementation

5.1 Software setup

The following software needs to be installed or downloaded before the development of the door label detection system.

1. Anaconda Navigator 3
2. Jupyter Notebook
3. Tesseract OCR V5.3.3

Several libraries and packages need to be installed before running the system. The table below shows the command line to install the libraries or packages using the Anaconda command prompt.

Package / Library	**Command Line**
Python Tesseract OCR	pip install pytesseract
Google Text-to-Speech	pip install gtts
Mutagen	pip install mutagen
FuzzyWuzzy	pip install fuzzywuzzy
Python Speech Recognition	pip install SpeechRecognition
Pyaudio	pip install pyaudio

Table 5.1.1 Command pipeline

Python

Python is a high-level programming language that is famous and commonly used around the world. It has a high-level built-in data structure and is often used as a glue language to connect components. Python has a huge collection of libraries that users can work with. The libraries needed in this project such as OpenCV, pyTesseract ORC, and Google Text-to-Speech library can be run on Python programming languages.

Anaconda

Anaconda is an open-source platform that allows us to write and execute Python code without any subscription or payment needed. Anaconda Python has many features and built-in libraries that we can use in this project.

Jupyter Notebook

Jupyter Notebook was created by a nonprofit organization to develop open-source software, open standards, and services for interactive computing across programming languages. It is suitable for performing tasks such as machine learning, deep learning, statistical modeling, and tasks that are related to data science.

Tesseract OCR

Tesseract is an open source Optical Character Recognition engine that can be directly used or used as an application programming interface (API) to extract text from images. Tesseract OCR uses a famous deep learning network which is the Convolutional Neural Network (CNN) to predict each of the characters in the image. There are a total of 14 page segmentation modes (PSMs) for a user to choose that best fit for their purpose. The PSMs used in this project is mode 6.

Google Text-to-Speech (gTTS)

gTTS is a Python library and command line interface (CLI) tool to interact with Google text-to-speech translation API. It can convert words or sentences into spoken MP3 files for further use. It supports many languages such as Chinese, English, Tamil, and more.

Google Speech-to-Text

Google speech-to-text is the service provided by the Google Cloud service for converting the mp3 file into text. With the help of the Python library speech recognizer, it will help to send the mp3 file to the Google Cloud server through the API. Google Speech to Text supports 125 languages for translation and conversion.

FuzzyWuzzy

FuzzyWuzzy is a Python library that compares two different string values by using the Lavenshtien distance method to calculate the differences between sequences and patterns. It is free to use because FuzzyWuzzy is an open-source library that developed by SeatGeek.

Pyaudio

Pyaudio is a library in Python that manages the input/output operation of the audio in the computer. It makes a more easier connection between the computer and the system to perform recording and playing audio.

5.2 Hardware Setup

The hardware involved in this project is a laptop, a webcam, and an earphone. The laptop will be used to run the image processing, door label reading, and pathfinding algorithms and convert the text into speech. The web camera acts as the user's eyes to capture the image of the environment and send it to the laptop for processing. The earphone is used to provide voice information to the user after the instruction is produced on the laptop.

Figure 5.2.1 Acer Swift 3 Laptop

Description	Specifications
Model	Acer Swift 3
Processor	11th Gen Intel(R) Core(TM) i5-1135G7
Operating System	Windows 11

Graphic	Intel® Iris® Xe Graphics
Memory	8 GB
Storage	512 GB SSD

Table 5.2.1 Specifications of laptop

Figure 5.2.2 allParts Webcam for Laptop

Description	**Specification**
Model	allParts
Video Format	Audio Video Interleave (AVI)
Frame Rate	30FPS
Resolution	1920 x 1080 P
Output Format	M-JPEG
Focusing Range	20mm
Interface Type	USB2.0

Table 5.2.2 Specification of Webcam

Figure 5.2.3 Baseus Encok WM01 Earphone

Description	Specification
Model	Baseus Encok WM01
Connectivity	Wireless (Bluetooth V5.0)
Operating Range	Up to 10 meter
Frequency Range	20Hz to 20kHz
Battery Capacity	40mAh (earphone), 200mAh (charging case)

Table 5.2.3 Specification of Earphone

5.3 System Operation

5.3.1 System Camera Display

The result from the process will be shown on the video display when the system is executing. The system display will be shown below:

There are only three main outputs that we are concerned about which are the detected result, command, and the user destination. These three outputs will be always shown at the bottom left corner of the display screen to let us observe the correctness of the input and the output. Figure 5.3.1 displays the idle system which did not perform any operation.

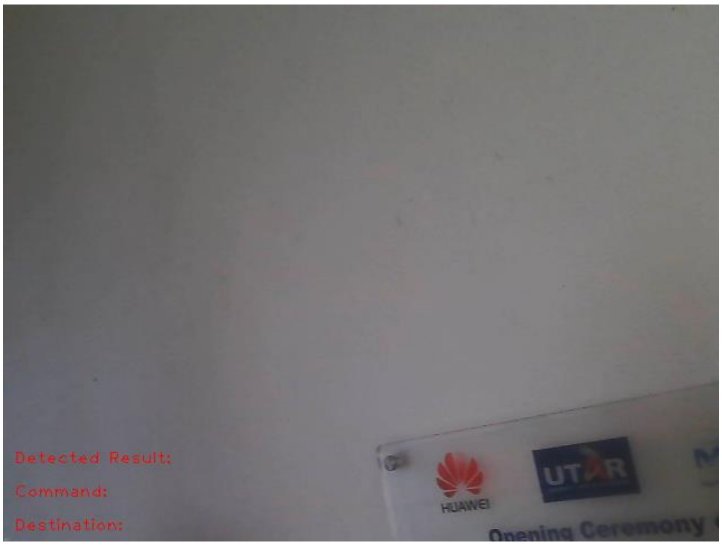


Figure 5.3.1 System display when idle

When the user does not input their destination, the system will only voice output the detected door label name to the user. The information shown in the display is just only the detected result. Figure 5.3.2 shows that the system will only voice out the door label name to the user.

Figure 5.3.2 System display when no user destination

When the user inputs their destination, the system will generate the guiding instructions for the user. Then, the Command will show the guiding instructions and the destination will be shown at the bottom (Figure 5.3.3). The system will only voice out the command to the user since it is in a guiding mode.

Figure 5.3.3 System display when received user destination

When the user successfully reaches the destination, the system will tell the user by voicing out the door label name to the user (Figure 5.3.4). Therefore, the user is able to know that they have reached their destination immediately.

Figure 5.3.4 System display when reaching the destination

5.4 Implementation Challenges and Issues

The first challenge faced while implementing the system is the real-time issues. The system is running smoothly when the captured image does not contain any red dot because we set the system to only run the OCR when there is a red dot. Therefore, when a red dot appears in the captured image, the system will be performing OCR and other processes. Therefore, the time required to process one single frame will be around 0.6 to 0.8 seconds delay per frame. The solution we proposed was to only perform the OCR every 0.5 seconds to solve the delaying problem. The outcome of this solution helps to reduce a lot of the delaying problems compared to the system implemented in FYP1 but there is still a little bit of delay happen.

The second challenge was the OCR performance. The Tesseract OCR can only detect the word in white color while recognizing the words on the door label. As we know, the door label in the building of FICT was divided into two parts which are the room code and room name. room code was in black colour font and the room name was in the white colour font. The OCR mostly detects only the door name but not the door code. For example, the N101 Lecture Room 1 door label was sent to the Tesseract OCR and the output was only “Lecture Room 1” and the “N101” was not able to be recognized by the OCR. The proposed solution for this is to create a look-up list that stores all the door label names and codes. Once we perform the OCR, we look up the list to get the full name and code of the list and return it to the user.

CHAPTER 6

System Evaluation and Discussion

6.1 System Testing Result

6.1 Red Dot Detection

Red dot detection plays an important role in this project since no reading will be performed unless a red dot is detected. Therefore, we need to know the limit of the red dot detection so that we can further enhance the experience of the user while using this system. Environment will be a crucial factor that can affect the performance of the red dot detection. This is because this red colour dot detection is extremely sensitive to colours. Environments such as rainy days or nights which will make the environment darker will affect the detection performance badly. Multiple red dots will also be a challenge for the system to locate the position of the door label. If multiple red dots do appear in one single frame, the system will have no choice but to detect all the red dots and perform OCR to prevent the system from missing out on the red dots that are located on the door label.

Besides environmental factors, user behavior is also one of the factors that may affect the red dot detection performance. This is because while the user is moving, the motion of the user will affect the image capture performance of the webcam and cause the captured image is blur. After that, the distance between the user and the door label will also affect the detection of the red colour dot. Therefore, it is important to know what the maximum range is for the system can detect the red colour dot successfully.

6.1.1 Common Case

The first environment to test the system is in the UTAR FICT block with sufficient light conditions in the building. This environment is usually during the daytime with sunny weather that can provide the whole building a sufficient sunlight and minimize the places that are very low in light conditions. The reason for choosing this as a testing environment is because this scenario commonly happened in the UTAR FICT block. Observing how well the red dot detection performs in a usual case scenario can let us identify the common issues that user might face during their regular use.

The testing result (Figure 6.1.1) was going well that all the red colour dots were able to be detected under a stable condition which is the image captured was firm and clear. This is because, with a sufficient amount of light provided by the environment, the red colour can easily be separated from the system. Therefore, the system works well in an environment that has a sufficient amount of light.

Original Image	Red dot detection
	
	
	

Table 6.1.1 Red dot normal test case

6.1.2 Dark Environment Case

The next scenario to test the system is in a dark place. The dark environment could be a rainy day or nighttime and no light is turned on in the UTAR FICT block. Unfortunately, we have no data for rainy days because we need to attend classes and miss out on the chance to get the data. The reason for choosing these environments as testing environments is that we need to determine the system limits under a limited intensity of light. In a real-world scenario, the condition of the light could vary from time to time. By observing how the red dot detection runs in a low light condition, we can evaluate the system's ability to by how well it can detect and locate the red colour dot in a real-world scenario.

The first environment is the light bulb of the walking pathway was faulty in the afternoon (Figure 6.1.1). The detection goes well since there is sunlight provides a little brightness to the captured frame hence the system is still able to differentiate the red colour dot out from the background. The second environment is testing in the nighttime of the UTAR FICT block. The detection also went well because the light bulb in the FICT block was strong enough to provide sufficient brightness for the detection. The test results are shown in Table 6.1.2 and Table 6.1.3.

Figure 6.1.1 Pathway with broken lightbulb

Original Image	**Red dot detection**
	

Table 6.1.2 Red dot faulty lightbulb case

Original Image	**Red dot Detection**
	
	
	

Table 6.1.3 Red dot nighttime test case

6.1.3 Multiple Red Dots

The third scenario is when there are multiple red dots appear in a single frame. This testing is to find out how the system will react to the frame that has multiple red dots around the door label. As the result shown in Table 6.1.4, if the red dots are near each other, there are still some red dots that are not detected. This is because the system will only detect one red dot in a certain radius around the detected red dot. For example, if there are three red dots detected in the radius of 5, the system will only detect the red dot located in the middle among them. With the help of this algorithm, the burden of segmentation and OCR reading will reduce and enhance the performance speed of the system.

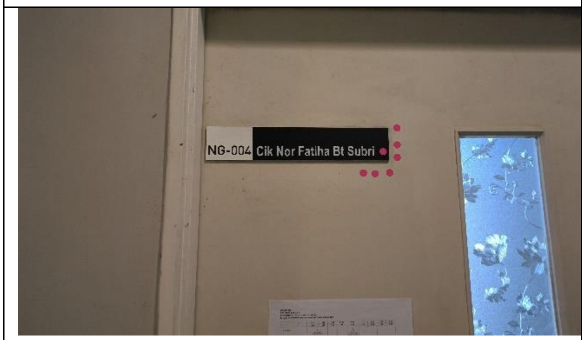
Original Image	Red dot detection
	
	
	

Table 6.1.4 Multiple red dot test case

6.1.4 Distance Testing

The fourth case to do testing is what is the maximum range the red dot can be detected between the user and the door label. In other words, how small is the red dot not able to be detected by the system? The OCR will also be testing to know the limit of how small can a character be to perform a success recognition and the results are shown in Table 6.1.5.

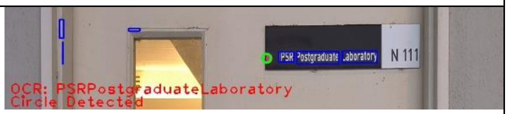
Distance / Cases	**Input**	**Output / Explanation**
2m		 The red colour dot is able to detect but the OCR cannot recognize that there are any words in the image because the size is too small.
1.5m		 The red colour dot is able to detect and the OCR can recognize the location of the words on the door label.
1m		 The red colour dot is able to detect and the OCR can recognize the location of the words on the door label.

Table 6.1.5 Red dot range test case

6.2 Ocr Detection

Motion Testing

For OCR detection, the concern is how blurry the door label can be read by OCR. This is because the walking motion of the user will cause the camera to shake and lead to a blurred image captured. The table below shows the captured frame of the door label when the user is walking. The frame is shown one by one to let us observe how blur the OCR to performs recognition well on the door label. The first two frames are captured when the user is walking at a normal movement speed. AS the result shown in Table 6.2.1, the image is a bit blurry and this causes the OCR not functioning. But when the user slows down due to the system notification, the OCR starts to perform well as the image gets clearer. The results are shown in table 6.2.1.

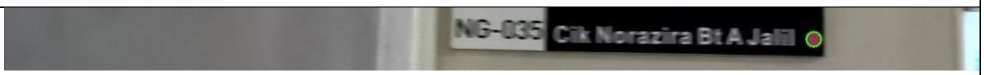
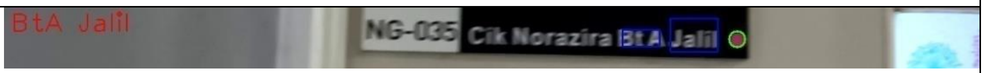
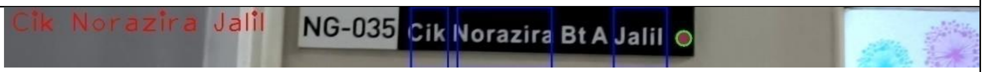
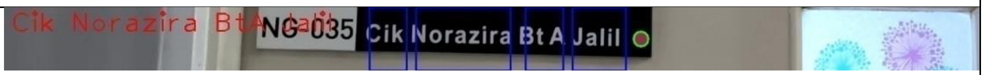
Speed	Captured frame
Normal	
Normal	
Slow	
Slow	
Stop	

Table 6.2.1 OCR blurring test case

6.3 Navigation Testing

For the navigation, a shortest path is generated based on the algorithm and the system will start to generate a command to the user based on the detected door label name. We keep the command as simple as it can to avoid the user getting confused while navigating inside the building. The location of the user and the destination are shown in Figure 6.3.1.

First of all, we need to receive the user input, After the user input, the system will mark down the user's current location and the destination node. Then, the shortest path algorithm will be generated. Once the door label is detected, the system will update the graph map and generate the guiding instructions. In the case below, the system tells the user to walk forward. The screen display is shown in Figure 6.3.2.

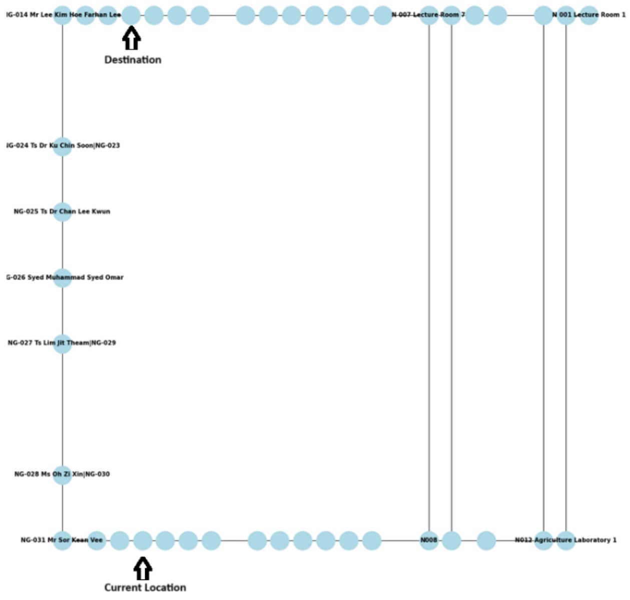


Figure 6.3.1 Graph map of the FICT

Figure 6.3.2 Navigation Display Screen

When the user reaches the end of the corridor, the system will generate the command to guide the user to turn right or turn left. Figure 6.3.3 shows the updated graph map and Figure 6.3.4 shows the command to tell the user to turn right.

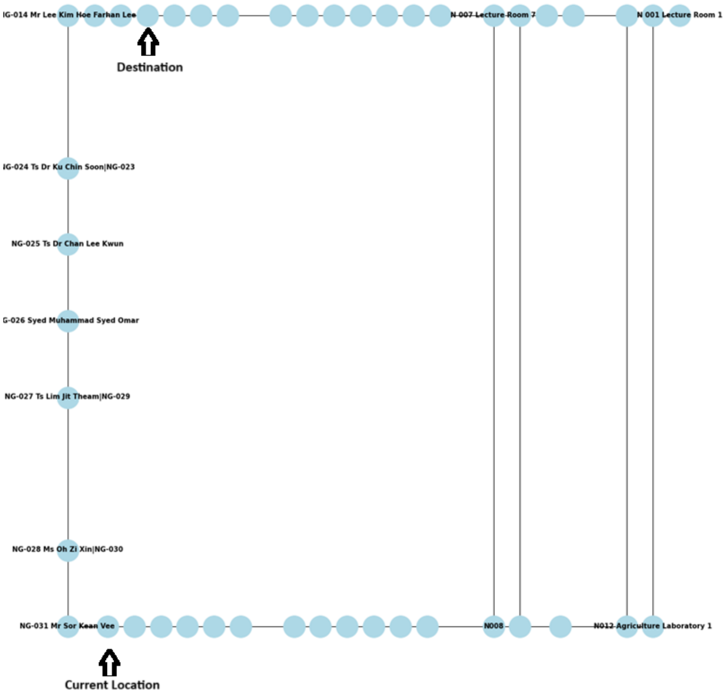


Figure 6.3.3 Updated Graph map

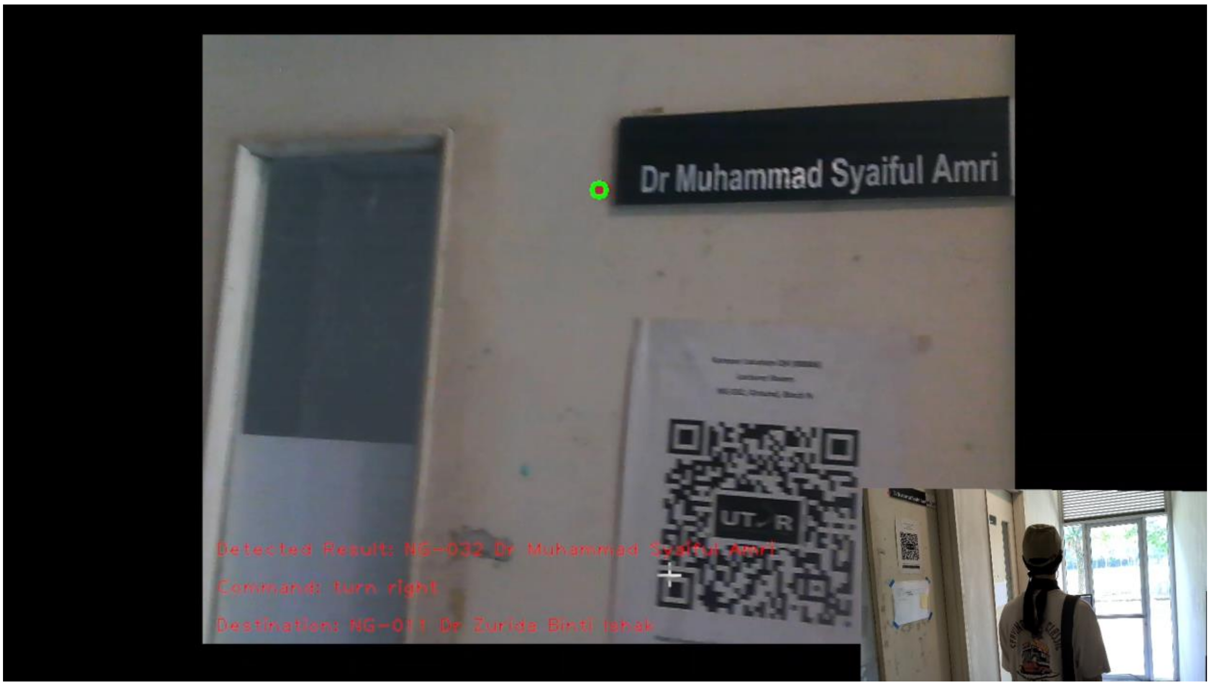


Figure 6.3.4 Navigation turn right

Once the user reaches the destination, the system will tell the user that he/she has reached the destination when the Camera detects the destination door label. Figure 6.3.5 shows the updated graph map and Figure 6.3.6 shows the navigation command given to the user when reaching the destination.

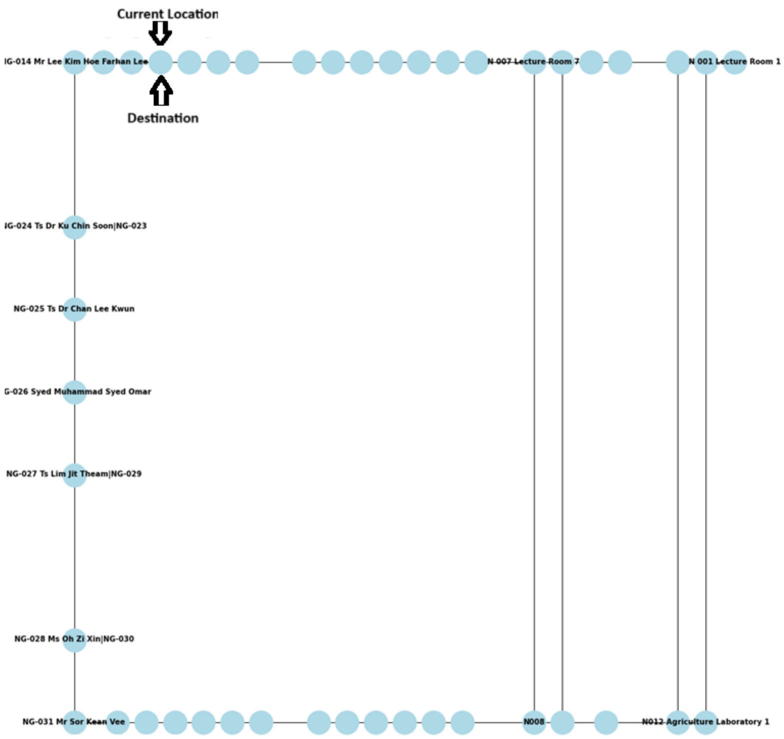


Figure 6.3.5 Updated graph map 2

Figure 6.3.6 Navigation reached destination

The system can only detect the environment of the user left hand side because there is only one web camera is used in this project. Therefore, there is a case in which the destination is located on the right hand side, the system will tell the user that the destination will be on the right hand side of the user. Figure 6.3.7 shows the command to tell the user to turn around to go to their destination.

Figure 6.3.7 Navigation reached destination by turning right hand side

6.4 Voice Receiving Testing

When the guiding button is pressed by the user, the system will ask the user about their destination (Figure 3.6.1 line 1). After finishing the output of the question, the system will start to listen to the user's voice (Figure 3.6.1 line 2). After the voice is recorded, the system will send the recorded mp3 file to the Google Speech to Text service inside the Google Cloud through the API (Figure 3.6.1 line 3). After Google Text to Speech finishes the recognition, it will return the recognized user voice as a string data type (Figure 3.6.1 line 4). Then, the system will look up the most similar name in the graph map and set it as the user destination. When all the setup is completed, the system will be ready to guide the user from their current location to their destination.

```
Hi! Where do you want to go?
Listening...
Recognizing...
You said: Ts Tan Tiek Bon
Destination: NF-037 Ts Tan TiekBoon
You are all set.
```

Figure 6.4.1 Voice conversion process

CHAPTER 7

Conclusion and Recommendations

7.1 Conclusion

In conclusion, the outcome of this project was a system that is able to detect the door label and read the door label. Besides that, the system will also be able to guide the user from point A to point B by concurrently updating the user's current location every time a new door label is read by the system. The reading of the door label was done by using image processing techniques and the help of the Tesseract OCR. The map that needs to be built in the system was done by a graph with the help of data structure and algorithms technique and networkX. The instructions for the route navigation to the user were done by using Dijkstra's algorithm on the pre-build map. The user input command can be received by the system using a speech-to-text algorithm, and the navigation instructions are able to output to the user by text-to-speech algorithm.

7.2 Recommendations

There are several limitations of this system that can be improved in the future. The system can be further improved with the recommendations stated below:

1. Train the Tesseract OCR model to be able to detect handwriting so that the system can detect the door label that is created temporarily to handle some emergency cases.
2. Build a file-sharing cloud platform that stores all the prebuild map settings so that the system can work in many different buildings without needing to set up the environment again for each device.
3. Embed the system with obstacle detection so that it can provide a more precise walking instruction to guide the user traveling in the building.
4. Add a flashlight on the camera and turn the flashlight on when the average pixel value in the frame image has reached a certain threshold that determines the environment is dark.
5. Add on another camera so that the the system can process the visual information of the user user in their left and right hand side.

REFERENCES

[1] "Vision Impairment and blindness," World Health Organization, https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment (accessed Sep. 11, 2023).

[2] T. Schwarze, M Lauer, M Schwaab, M Romanovas, S Bohm and T Jurgensohn., "A camera-based mobility aid for visually impaired people," *KI - Künstliche Intelligenz*, vol. 30, no. 1, pp. 29–36, 2015. doi:10.1007/s13218-015-0407-7

[3] K. Chaccour and G. Badr, "Novel Indoor Navigation System for Visually Impaired and blind people," *International Conference on Applied Research in Computer Science and Engineering (ICAR),* 2015. doi:10.1109/arcse.2015.7338143

[4] J. Bai, S. Lian, Z. Liu, K. Wang, and D. Liu, "Virtual-blind-road following-based wearable navigation device for blind people," *IEEE Transactions on Consumer Electronics,* vol. 64, no. 1, pp. 136–143, 2018. doi:10.1109/tce.2018.2812498

[5] A. Ali and M. A. Ali, "Blind Navigation System for visually impaired using windowing-based mean on Microsoft Kinect camera," *Fourth International Conference on Advances in Biomedical Engineering (ICABME)*, 2017. doi:10.1109/icabme.2017.8167560

[6] C. Ton *et al.*, "Lidar assist spatial sensing for the visually impaired and performance analysis," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 9, pp. 1727–1734, 2018. doi:10.1109/tnsre.2018.2859800

[7] H. Jiang, T. Gonnot, W.-J. Yi, and J. Saniie, "Computer Vision and text recognition for assisting visually impaired people using Android smartphone," *IEEE International Conference on Electro Information Technology (EIT),* 2017. doi:10.1109/eit.2017.8053384

[8] R. C. Joshi, S. Yadav, M. K. Dutta, and C. M. Travieso-Gonzalez, "Efficient Multi-Object detection and smart navigation using artificial intelligence for visually impaired people," *Entropy,* vol. 22, no. 9, p. 941, 2020. doi:10.3390/e22090941

[9] M. Mukhiddinov and J. Cho, "Smart glass system using Deep Learning for the blind and visually impaired," *Electronics*, vol. 10, no. 22, p. 2756, 2021. doi:10.3390/electronics10222756

[10] A. Meliones and D. Sampson, "Blind museumtourer: A system for self-guided tours in museums and Blind Indoor Navigation," *Technologies*, vol. 6, no. 1, p. 4, 2018. doi:10.3390/technologies6010004

[11] B. Kuriakose, R. Shrestha, and F. E. Sandnes, "Tools and technologies for blind and visually impaired navigation support: A review," *IETE Technical Review*, vol. 39, no. 1, pp. 3–18, 2020. doi:10.1080/02564602.2020.1819893

[12] C. Wickramaarachchi *et al.*, “Can visual impairment impact your income potential?,” *PLOS ONE*, vol. 18, no. 4, 2023. doi:10.1371/journal.pone.0284553

APPENDIX A: WEEKLY LOG

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: Y3T3	Study week no.: Week 4
Student Name & ID: Kuan Wei Yeow 20ACB03477	
Supervisor: Prof. Dr Leung Kar Hang	
Project Title: Indoor Navigation for the Visually Impaired by Reading Door Label	

1. WORK DONE

Not much has done because I'm still improving the accuracy of the red dot detection and Tesseract OCR.

2. WORK TO BE DONE

1. Start to study about how to implement a graph to build the building floor plan map.
2. Ensure the detection of Tesseract OCR is able to scan the door label correctly.

3. PROBLEMS ENCOUNTERED

For (1), show captured images to convince other. On the other hand, if human cannot do it, it is OK for your work not able to do it too.

1. When the camera faced the sunlight (example: the windows), the camera will not be able to detect the door label in the next few seconds (exposure).
2. The circle detection of my system is using HoughCircle in OpenCV, but due to the motion the red dot circle cannot be detect easily. User needs to walk slowly to stable the camera to capture the door label. This is same for the OCR detection, if walk faster then the words will be blur.

The system should first detect red color and allert user of the red location. It then expects the user to slow down to allow clear image for further analysis.

3. Looking a way to convert the door label into black color and the background is white (I searched online they said it will improve the accuracy of the OCR)

The answer can easily be found from Internet. If cannot, then come to me.

4. SELF EVALUATION OF THE PROGRESS

The progress is slow because I still enhancing the previous feature.

Supervisor's signature

25 Feb 2024

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: Y3T3	**Study week no.: Week 6**
Student Name & ID: Kuan Wei Yeow 20ACB03477	
Supervisor: Prof. Dr Leung Kar Hang	
Project Title: Indoor Navigation for the Visually Impaired by Reading Door Label	

1. WORK DONE

Look back all the method I used to ensure it won't affect the text in the image to be scan in Tesseract OCR.

2. WORK TO BE DONE

1. Study about how to implement a graph to build the building floor plan map.
2. Study how to use the Google speech to text and text to speech to guild the user.

3. PROBLEMS ENCOUNTERED

1. I find out that binary threshold value will affect the reading of OCR.

Threshold 30

But when the place is brighter, the threshold 30 is not usable.

Can try out not doing thresholding at all and check the results by sending to Tesseract (1) the original cut out image, (2) the image after edge detection.

So, I'm thinking to find 3 type of threshold and send to the OCR at the same time so that this system is not afraid of brighter place or dimmer place.

4. SELF EVALUATION OF THE PROGRESS

No progress because I'm sick this week. Will try to catch up all the progress in week 7 and hope that I can start implement the floor plan map week 8.

Supervisor's signature

10 Mar 2024

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: Y3T3	**Study week no.: Week 7**
Student Name & ID: Kuan Wei Yeow 20ACB03477	
Supervisor: Prof. Dr Leung Kar Hang	
Project Title: Indoor Navigation for the Visually Impaired by Reading Door Label	

1. WORK DONE

Test run my system to found out the accuracy of red color dot detection and OCR performance. The accuracy of red dot detection is 70% and for OCR is 90%.

2. WORK TO BE DONE

1. Implement a graph to build the building floor plan map.
2. Study how to use the Google speech to text and text to speech to guild the user.

3. PROBLEMS ENCOUNTERED

1. The red color dot detection need to be further improved as few dots did not detect by the system. This is because some place is bright, and some place is dark (It was raining at that moment). In the darker place, the dot I placed on the door label is not able to detect by the system. It was only detectable when I use my phone's torchlight to make the place brighter. Cannot suggest anything unless the detection details are presented in your weekly report using a few pages. You should start giving detail like the system block diagrams.

2. The system needs the user to walk extremely slow or stand still to be able to recognize the words on the door label in the OCR, so I plan to do a function to notice the user slow down once the system detected red dot.

4. SELF EVALUATION OF THE PROGRESS

After testing my system, I need to fine tune my red dot detection to not to miss out all the red dots. I'll start implement the graph map and try the text to speech algorithm so that I know how many second the system needs to inform the position of the user. If it needs a longer time to inform user, then I'll suggest the user to stand still once the door label is detected.

Supervisor's signature
17 Mar 2024

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: Y3T3	**Study week no.: Week 8**
Student Name & ID: Kuan Wei Yeow 20ACB03477	
Supervisor: Prof. Dr Leung Kar Hang	
Project Title: Indoor Navigation for the Visually Impaired by Reading Door Label	

1. WORK DONE

Floor map of FICT building has completed using graph method.

2. WORK TO BE DONE

1. Implement an instruction generator to guild the user to his/her destination by receiving their destination.
2. Implement Google speech to text and text to speech to guild the user.

3. PROBLEMS ENCOUNTERED

The accuracy of using this system in lecturers' room is high, but when it goes to the scan the Lecture rooms, it is not so good. This is because my OCR cannot detect the full word on the door label all the time, so I used the words detected to look up in a pre-created list that store all the door label name in FICT.

For example, the first line is what my OCR detect from the door label, and the second line is the result get from the door label list by comparison method. This works fine for every lecturer's room and labs in the FICT. Possible questions from examiners could be (1) "what if the original texts are in black while the background is white?", and (2) What comparision method.

But for the lecture room, most of the time the OCR only captured the 'Lecture Room 5' and didn't capture the 'N 105', The system will not able to differentiate whether it is the ground floor 'N005 Lecture Room 5' or is the 1st floor 'N105 Lecture Room 5'.

Need to fix such problem. This is the most important part of your work. Any clue on why so?

Only detection that able to scan all the words can read the door label correctly. But only 1 out of 10 doors label the OCR will detect the full words on the door label.

Sound like your guess? This could be the examiner viewpoint unless you have done enough investigation to convince the examiners to be on your side.

I had check that the Tesseract OCR is the latest version and tested many of the configuration for the OCR. The best configuration so far, I get is it can detect the full door label 1 out of 10.
4. SELF EVALUATION OF THE PROGRESS When I completed the function that generate instruction to guide the user from A to B, this system is mostly completed. The things after that are to tuning for increase the color detection and increase the accuracy of reading the door label. I will start to write the report once all the function is implemented.

Supervisor's signature
24 March 2024

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: Y3T3	Study week no.: Week 9
Student Name & ID: Kuan Wei Yeow 20ACB03477	
Supervisor: Prof. Dr Leung Kar Hang	
Project Title: Indoor Navigation for the Visually Impaired by Reading Door Label	

1. WORK DONE

Floor map of FICT building has completed using graph method.

2. WORK TO BE DONE

1. Implement an instruction generator to guild the user to his/her destination by receiving their destination.
2. Implement Google speech to text and text to speech to guild the user.
3. Report writing.

3. PROBLEMS ENCOUNTERED

No problem encountered this week.

4. SELF EVALUATION OF THE PROGRESS

I had only little progress because of many midterms and practical test this week. Will catch up in next week. Look forward to this

Supervisor's signature
31 March 2024

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / Project II)

Trimester, Year: Y3T3	**Study week no.: Week 10**
Student Name & ID: Kuan Wei Yeow 20ACB03477	
Supervisor: Prof. Dr Leung Kar Hang	
Project Title: Indoor Navigation for the Visually Impaired by Reading Door Label	

1. WORK DONE

Floor map of FICT building has completed using graph method.
The instruction generator that guild user from point A to point B is completed.

Can show demo video?

2. WORK TO BE DONE

1. Implement Google speech to text and text to speech to guild the user.
2. Test the full system in week 11.
3. Report writing.
4. Find that there are any OCR will perform better than current OCR.

3. PROBLEMS ENCOUNTERED

No problem encountered this week.

4. SELF EVALUATION OF THE PROGRESS

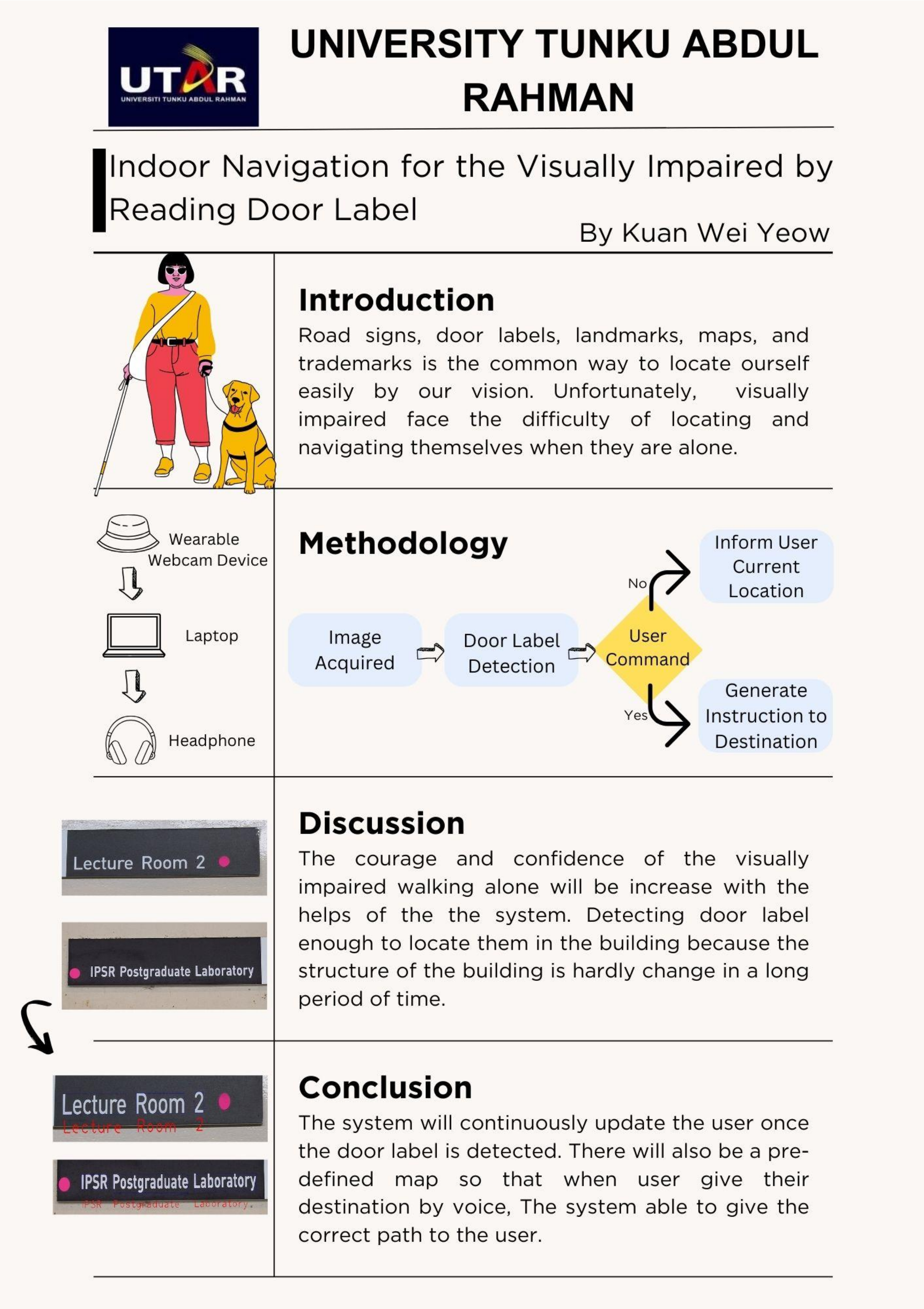
The system is almost done. I will test the system by blindfolding myself walking from point A to point B in FICT next week.

Supervisor's signature

7 Apr 2024

Student's signature

APPENDIX B: POSTER



PLAGIARISM CHECK RESULT

Indoor Navigation for the Visually Impaired by Reading Door Label

ORIGINALITY REPORT

7%	4%	4%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.utar.edu.my Internet Source	2%
2	Jinqiang Bai, Shiguo Lian, Zhaoxiang Liu, Kai Wang, Dijun Liu. "Virtual-Blind-Road Following-Based Wearable Navigation Device for Blind People", IEEE Transactions on Consumer Electronics, 2018 Publication	1%
3	Mukhriddin Mukhiddinov, Jinsoo Cho. "Smart Glass System Using Deep Learning for the Blind and Visually Impaired", Electronics, 2021 Publication	1%
4	Tobias Schwarze, Martin Lauer, Manuel Schwaab, Michailas Romanovas, Sandra Böhm, Thomas Jürgensohn. "A Camera-Based Mobility Aid for Visually Impaired People", KI - Künstliche Intelligenz, 2015 Publication	<1%
5	ijirset.com Internet Source	<1%

6	Submitted to Universiti Tunku Abdul Rahman Student Paper	<1%
7	Kabalan Chaccour, Georges Badr. "Computer vision guidance system for indoor navigation of visually impaired people", 2016 IEEE 8th International Conference on Intelligent Systems (IS), 2016 Publication	<1%
8	Michał Styła, Przemysław Adamkiewicz. "OPTIMISATION OF COMMERCIAL BUILDING MANAGEMENT PROCESSES USING USER BEHAVIOUR ANALYSIS SYSTEMS SUPPORTED BY COMPUTATIONAL INTELLIGENCE AND RTI", Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska, 2022 Publication	<1%
9	Submitted to De La Salle University Student Paper	<1%
10	www.cv-foundation.org Internet Source	<1%
11	Apostolos Meliones, Demetrios Sampson. "Blind MuseumTourer: A System for Self-Guided Tours in Museums and Blind Indoor Navigation", Technologies, 2018 Publication	<1%
12	Rakesh Chandra Joshi, Divyanshu Singh, Vaibhav Tiwari, Malay Kishore Dutta. "An	<1%

efficient deep neural network based abnormality detection and multi-class breast tumor classification", Multimedia Tools and Applications, 2021
Publication

13	D.R. Vedhaviyassh, R. Sudhan, G. Saranya, M. Safa, D. Arun. "Comparative Analysis of EasyOCR and TesseractOCR for Automatic License Plate Recognition using Deep Learning Algorithm", 2022 6th International Conference on Electronics, Communication and Aerospace Technology, 2022 Publication	<1 %
14	Submitted to Sim University Student Paper	<1 %
15	Durga Shanmukhi Kunapareddy, Naga Pavan Kumar Putta, Viswanath Anand Maddala, Ranjith Kumar Bethapudi, Siva Reddy Vanga. "Smart Vision based Assistant for Visually Impaired", 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), 2022 Publication	<1 %
16	Submitted to RMIT University Student Paper	<1 %
17	123dok.com Internet Source	<1 %

18	Fazliddin Makhmudov, Mukhriddin Mukhiddinov, Akmalbek Abdusalomov, Kuldoshbay Avazov, Utkir Khamdamov, Young Im Cho. "Improvement of the End-to-End Scene Text Recognition Method for "Text to Speech" Conversion", International Journal of Wavelets, Multiresolution and Information Processing, 2020 Publication	<1 %
19	Submitted to Morgan Park High School Student Paper	<1 %
20	research.sabanciuniv.edu Internet Source	<1 %
21	Submitted to City University of Hong Kong Student Paper	<1 %
22	Jayakanth Kunhoth, AbdelGhani Karkar, Somaya Al-Maadeed, Abdulla Al-Ali. "Indoor positioning and wayfinding systems: a survey", Human-centric Computing and Information Sciences, 2020 Publication	<1 %
23	docshare.tips Internet Source	<1 %
24	Kris Luyten, Kristof Verpoorten, Karin Coninx. "Ad-hoc co-located collaborative work with mobile devices", Proceedings of the 9th	<1 %

	international conference on Human computer interaction with mobile devices and services, 2007 Publication	
25	ilocos.da.gov.ph Internet Source	<1%
26	Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, Sergey Zagoruyko. "Chapter 13 End-to-End Object Detection with Transformers", Springer Science and Business Media LLC, 2020 Publication	<1%
27	pure.tue.nl Internet Source	<1%
28	www.farnell.com Internet Source	<1%
29	access.ex.ac.uk Internet Source	<1%
30	archive.org Internet Source	<1%
31	idoc.pub Internet Source	<1%
32	neptune.ai Internet Source	<1%

33	HANHWE KIM, STEPHEN C. HIRTLE. "Spatial metaphors and disorientation in hypertext browsing", Behaviour & Information Technology, 2007 Publication	<1 %
34	Jinqiang Bai, Shiguo Lian, Zhaoxiang Liu, Kai Wang, Dijun Liu. "Virtual-Blind-Road Following Based Wearable Navigation Device for Blind People", IEEE Transactions on Consumer Electronics, 2018 Publication	<1 %
35	Tobias Schwarze, Martin Lauer, Manuel Schwaab, Michailas Romanovas, Sandra Bohm, Thomas Jurgensohn. "An Intuitive Mobility Aid for Visually Impaired People Based on Stereo Vision", 2015 IEEE International Conference on Computer Vision Workshop (ICCVW), 2015 Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

Universiti Tunku Abdul Rahman			
Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

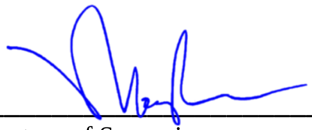
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Kuan Wei Yeow
ID Number(s)	20ACB03477
Programme / Course	Bachelor of Computer Science (Honours)
Title of Final Year Project	Indoor Navigation for the Visually Impaired by Reading Door Label

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)**
Overall similarity index: __7_____ **%** **Similarity by source** Internet Sources: ____4___ % Publications: ____4___ % Student Papers: ____2___ %	
Number of individual sources listed of more than 3% similarity: _0________	
Parameters of originality required, and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note: Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.



____
Signature of Supervisor

Name: Leung Kar Hang

Date: 25 April 2024

____
Signature of Co-Supervisor

Name: ________________________

Date: ________________________

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	20ACB03477
Student Name	KUAN WEI YEOW
Supervisor Name	PROF. DR LEUNG KAR HANG

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
/	Title Page
/	Signed Report Status Declaration Form
/	Signed FYP Thesis Submission Form
/	Signed form of the Declaration of Originality
/	Acknowledgement
/	Abstract
/	Table of Contents
/	List of Figures (if applicable)
/	List of Tables (if applicable)
	List of Symbols (if applicable)
/	List of Abbreviations (if applicable)
/	Chapters / Content
/	Bibliography (or References)
/	All references in bibliography are cited in the thesis, especially in the chapter of literature review
	Appendices (if applicable)
/	Weekly Log
/	Poster
/	Signed Turnitin Report (Plagiarism Check Result – Form Number: FM-IAD-005)
/	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

kuan

____

(Signature of Student)
Date: 25 April 2024