MICROPROCESSOR-BASED ON-STREET PARKING NOTIFICATION SYSTEM

CHRISTINA KO YUN FAN

BACHELOR OF MECHATRONICS ENGINEERING WITH HONOURS
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

MICROPROCESSOR-BASED ON-STREET PARKING NOTIFICATION SYSTEM

CHRISTINA KO YUN FAN

A report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Mechatronics Engineering with Honours

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

DECLARATION

I declare that this thesis entitled “MICROPROCESSOR-BASED ON-STREET PARKING NOTIFICATION SYSTEM is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : ....

Name : ....

Date : ....

APPROVAL

I hereby declare that I have checked this report entitled "Microprocessor-based On-street Parking Notification System" and in my opinion, this thesis it complies the partial fulfillment for awarding the award of the degree of Bachelor of Mechatronics Engineering with Honours

Signature : ....

Supervisor Name : ....

Date : ....

DEDICATIONS

To my beloved mother and father

ACKNOWLEDGEMENTS

In preparing this report, I was in contact with many people, researchers, academicians and practitioners. They have contributed towards my understanding and thought. In particular, I wish to express my sincere appreciation to my main project supervisor, Professor Dr. Muhammad Herman bin Jamaluddin. His prompt encouragements, timely assistance, meticulous scrutiny, scholarly advice and warm kindness have thrust me beyond my boundaries in completion of my Final Year Project (FYP). I am very thankful to my panels who are Professor Madya Dr. Chong Shin Hong and Dr. Hyreil Anuar bin Kasdirin for their feedback and advices. Without their suggestions, this project would not have been same as presented here.

Besides, I am also indebted to the researches for their assistance in providing the relevant literatures and findings. Nevertheless, I would take this opportunity to express my gratitude to my parents and y family for their continuous shower of love, unceasing encouragement and support throughout all these years. My fellow friends should also be recognized for their support. Their views and tips are useful indeed. Last but not least, I place on record, my sense of gratitude to one and all who, directly or indirectly, have offered their helping hand upon the completion of this project.

ABSTRACT

Due to the flaw existed in public transportation system in Malaysia, most of the locals will rather use their own transportation to travel to the desired destination. Thus, this causes the number of vehicles in Malaysia to increase, which leads to finding a vacant parking lot is getting tougher especially in a crowded area. Thus, it causes traffic congestion and consumes additional time and fuel. Therefore, an efficient and user-friendly on-street parking system with the implementation of Internet of Things (IoT) and cloud computing was able to meet the deficiency of current parking system in Malaysia. The objectives of this project is to design and develop a smart parking system using computer vision algorithm and the performance of the parking vacancy detection system in terms of accuracy was evaluated. In addition, a mobile application was introduced to enable the drivers to check the status of availability of parking space at anywhere and anytime. The microprocessor-based on-street parking notification system consisted of two parts which are software and hardware. The prototype was made up of an Intel UP Squared board which acted as a server to integrate all the inputs and outputs, also a medium to process the captures video or images. An IP camera with a tripod stand was set up to capture sharper and clearer images or videos. On the other hand, the software used to develop this parking notification system is Android Studio and Firebase. After that, the image that has been captured by the IP camera was being analysed via OpenCV in Intel UP Squared board and then the data was being sent to the Firebase and stored as a real-time database. The evaluations on the performance of data synchronization system between OpenCV, Firebase and mobile application and parking vacancy detection system at 3 different type of parking and at 4 different environments were done. The results has proved that the data in the images were successfully transferred to the Firebase and mobile application. The results obtained also showed that the contour parking vacancy detection system and Haar Cascade parking vacancy detection system has achieved 100% of accuracy when tested at three different types of parking. However, the accuracy of Haar Cascade parking vacancy detection system was dropped from 100% to 25% when it is tested at the different environments.

ABSTRAK

Oleh disebabkan beberapa kekurangan wujud dalam sistem pengangkutan awam di Malaysia, kebanyakan penduduk tempatan lebih suka menggunakan pengangkutan mereka sendiri untuk pergi ke destinasi yang dikehendaki. Oleh itu, ini menyebabkan jumlah kenderaan di Malaysia meningkat and membawa kesan kepada tempat letak kereta kosong semakin sukar dicari terutamanya di kawasan yang sesak. Oleh itu, sistem parkir di jalan raya yang cekap dan mesra dengan konsep Internet Perkara (IoT) dan pengkomputeran awan dapat memenuhi kekurangan sistem letak kereta sedia ada di Malaysia. Objektif projek ini adalah untuk merekabentuk dan membangunkan sistem letak kereta pintar menggunakan algoritma penglihatan komputer dan prestasi sistem pengesanan kekosongan tempat letak kereta dari segi ketepatan. Di samping itu, aplikasi mudah alih diperkenalkan untuk membolehkan pemandu memeriksa status ruang letak kenderaan di mana-mana dan pada bila-bila masa. Sistem notis letak kereta di atas mikropemproses terdiri daripada dua bahagian iaitu perisian dan perkakasan. Prototaip ini terdiri daripada papan Intel Squared UP yang bertindak sebagai pelayan untuk mengintegrasikan semua input dan output, juga medium untuk memproses video atau gambar menangkap. Kamera IP dengan pendirian tripod telah disediakan untuk mengambil imej atau video yang lebih tajam dan jelas. Sebaliknya, perisian yang digunakan untuk membangunkan sistem pemberitahuan parkir ini adalah Android Studio dan Firebase. Selepas itu, imej yang telah ditangkap oleh kamera IP sedang dianalisa melalui OpenCV di papan Intel Squared UP dan kemudian data dihantar ke Firebase dan disimpan sebagai pangkalan data masa nyata. Penilaian mengenai prestasi sistem penyegerakan data antara sistem pengesanan OpenCV, Firebase dan mudah alih dan sistem pengesanan kekosongan tempat letak kereta di 3 jenis tempat letak kereta yang berbeza dan di 4 persekitaran yang berbeza telah dilakukan. Hasilnya telah membuktikan bahawa data dalam imej berjaya dipindahkan ke aplikasi Firebase dan mudah alih. Keputusan yang diperoleh menunjukkan bahawa sistem pengesanan kekosongan tempat letak kenderaan kontemporari dan sistem pengesanan kekosongan tempat letak kereta Haar Cascade telah mencapai 100% ketepatan apabila diuji di tiga jenis tempat letak kereta. Walau bagaimanapun, ketepatan sistem pengesanan kekosongan letak Haar Cascade digugurkan dari 100% ke 25% apabila ia diuji di persekitaran yang berbeza.

TABLE OF CONTENTS

	PAGE
DECLARATION	
APPROVAL	
DEDICATIONS	
ACKNOWLEDGEMENTS	**1**
ABSTRACT	**2**
ABSTRAK	**3**
TABLE OF CONTENTS	**4**
LIST OF TABLES	**6**
LIST OF FIGURES	**7**
LIST OF SYMBOLS AND ABBREVIATIONS	**8**
LIST OF APPENDICES	**9**
CHAPTER 1 INTRODUCTION	**10**
1.1 Introduction	10
1.2 Background of Study	10
1.3 Motivation	11
1.4 Problem Statement	12
1.5 Objectives of Project	13
1.6 Scope of Project	13
1.7 Thesis Outline	14
CHAPTER 2 LITERATURE REVIEW	**15**
2.1 Introduction	15
2.2 Effective Parking Management System	15
2.3 Comparison of Existing Open Space Parking System	16
2.4 Comparison of the Computer Vision Algorithm Used for Detecting Car Parks	21
2.5 Summary	23
CHAPTER 3 METHODOLOGY	**24**
3.1 Introduction	24
3.2 System Overview	25
3.3 Development of Microprocessor-based On-street Parking System	27
3.3.1 Design of Parking Vacancy Detection System	27
3.3.2 Design of Mobile Application	31
3.3.3 Design of the Data Synchronization System Using IoT Ecosystem	34
3.4 Experiment Design	37
3.4.1 Evaluation on the Performance of Data Synchronization System	37

3.4.2 Evaluation on the Performance of Parking Vacancy Detection System 38
3.5 Summary 42

CHAPTER 4 RESULTS AND DISCUSSIONS 43
4.1 Introduction 43
4.2 Results on the Integrations between Hardware and Software using IoT Ecosystem 43
4.3 Experiments on the Evaluations of the Accuracy of Parking Vacancy Detection System 46
4.3.1 Contour Parking Vacancy Detection with Different Settings of Radius Range 46
4.3.2 Comparison of the Performance between Contour Parking Vacancy Detection System and Haar Cascade Parking Vacancy Detection System at Different Types of Parking 49
4.3.3 Comparison of the Performance between Contour Parking Vacancy Detection System and Haar Cascade Parking Vacancy Detection System at Different Environments 54
4.4 Summary 59

CHAPTER 5 CONCLUSION AND RECOMMENDATIONS 60
5.1 Introduction 60
5.2 Conclusion 60
5.3 Recommendations 62

REFERENCES 63

APPENDICES 66

LIST OF TABLES

Table 2.1: The Comparison Table Between The Existing Open Space Parking Systems 16

Table 2.2: The Comparison Table of Computer Vision Algorithms Used for Parking Detection 21

Table 3.1: The Selected Components with Their Specifications 26

Table 4.1: Real-time Data Synchronization between OpenCV, Firebase and Mobile Application 44

Table 4.2: The Experiment on the Contour Parking Vacancy Detection System with Different Settings of Radius Range 47

Table 4.3: Parking Vacancy Detection System Using Contour-Detection Method and Haar Cascade Classifier at Perpendicular Parkings 49

Table 4.4: Parking Vacancy Detection System Using Contour Detection Method and Haar Cascade Classifier at Angle Parkings 51

Table 4.5: Parking Vacancy Detection System Using Contour Detection Method and Haar Cascade Classifier at Parallel Parkings 52

Table 4.6: Results of the Contour Parking Vacancy Detection System at Different Environment 54

Table 4.7: Results of the Haar Cascade Parking Vacancy Detection System at Different Environment 56

Table 4.8 : The Pros and Cons of Contour-detection method and Haar Cascade Classifier 58

Table 4.9 : Summary of Experiments 59

LIST OF FIGURES

Figure 2.1: The Basic Concept of An Effective Parking Managemnet System 16

Figure 3.1: System Overview of This Project 25

Figure 3.2: Flowchart of The Parking Vacancy Detection System with Contour-detection Method 28

Figure 3.3: Cascade Trainer GUI 29

Figure 3.4: Add Path of Sample Folder To Cascade Trainer GUI 30

Figure 3.5: The flows of Developing Haar Cascade Algorithm 31

Figure 3.6: Flowchart of Mobile Application 32

Figure 3.7: User Interfaces(UI) of Registration, Password Reset and Login Activity 33

Figure 3.8: User Interface of The Drawer of Parking Application 34

Figure 3.9: Steps to Integrate OpenCV and Firebase 35

Figure 3.10: Linking Approach Between Firebase and Android Application 36

Figure 3.11: The Observations from The Display Platforms 38

Figure 3.12: The Selection for The Most Preferred Radius Range 39

Figure 3.13: Experiment Setup for Software and Hardware 39

Figure 3.14: Settings of Parameter for the Experiments 40

Figure 3.15: Experiment Setup for Perpendicular Parking 40

Figure 3.16: Experiment Setup for Angle Parking 40

Figure 3.17: Experiment Setup for Parallel Parking 41

Figure 3.18: The Captured Images From the Reference Environment 41

LIST OF SYMBOLS AND ABBREVIATIONS

IoT	-	Internet of Things
OS	-	Operating System
PC	-	Personal Computer
UI	-	User Interface

LIST OF APPENDICES

APPENDIX A	GANTT CHART FOR FYP 1	66
APPENDIX B	GANTT CHART FOR FYP 2	67
APPENDIX C	COUNTOUR-DETECTION METHOD's CODE	68
APPENDIX D	HAAR CASCADE CLASSIFIER's CODE	70

CHAPTER 1

INTRODUCTION

1.1 Introduction

The current parking systems are mainly focused on the indoor parking system. However, the on-street parking system should be highlighted because most of the traffic issues such as traffic jam and double-parked are happened at outdoor. Therefore, this chapter highpoints the background of study, problem statement, objectives and scopes of the project. Background studies depicts the important of an effective parking system and the technologies used to be implemented in parking system. The issues and problems that are aroused due to the ineffective parking system in Malaysia are stated. Besides, the flaws of the current on-street parking system in Malaysia are also discussed as the problem statement of this project. To improve the existing on-street parking system, 2 objectives are suggested as a benchmark. Lastly, several scopes is mentioned to define the limits and boundaries of the project in overseeing the project upon completion.

1.2 Background of Study

Due to the increasing in population in the industrialization of the world, the mismanagement of the parking system to the available parking space leads in parking related problems. An effective parking system is needed to be developed in order to manage the available the parking lots and satisfy the parking lots demand. The word “effectiveness” provides different ideas and meanings based on the requirements that are needed in a system. There are many parking systems with different technologies are proposed by the researches from different countries in order to developed an effective parking system.

In [1], a survey about the various technologies and techniques for an intelligent car parking system is done by Faheem and his teams. The technologies and techniques that are commonly implemented in the parking system are wireless sensor based system, vision based system, expert system, GPS based system and so on. These systems are used to detect the parking availability at a parking area. In order to make the parking system more effective, the system always uses the concept of Internet of Things (IoT). IoT is introduced by Kevin Ashton in 1998 [2]. It is becoming so important to integrate with cloud computing because it provides a platform to store more data and develops smart applications for the users.

1.3 Motivation

Due to the ineffective parking system, the drivers need to spend more time and costs in searching a parking lot to park their vehicles. The ineffective parking system leads the double-parked problem and it causes the traffic congestion in the crowded area. This double-parked problem also causes the incidents happened due to conflict and misunderstanding between two parties. A real incident is happened at a restaurant in Bentong, Pahang. According to the article reported by China Press, the Toyota truck owner got angry and knocked the double-parked Honda City after he honked it for half an hour [3]. Besides, the double-parked problem also leads traffic congestion occurred at SS 15 Subang Jaya because the drivers do not want to waste their time in searching a vacant parking to park their car properly [4]. According to the survey done by Uber, a Malaysian spends average 25 minutes each day in looking for a parking spot and this causes 74% of them have missed or been late to work [5].

In addition, this ineffective parking system causes the burning of addition fuel in searching a parking space in crowded area causes the environment impacts. The carbon dioxide, CO_2 that is released when the burning of petrol causes the global warming problem. The global warming problem leads the temperature of the world changing dramatically and unexpectedly [6]. The burning of the petrol leads the problem of air pollution. Air pollution affects quality of air and causes the accidents occurred due to the thick fog formed by the burning of fuel. The air pollution also will harm the human’s health such as asthma problem and eye disease.

The development of the advanced technologies is one of the motivation for developing an effective parking system. Director Chong Kai Wooi explained that Malaysia is on the track of smart cities, paying parking using application via smartphone is a possible way to improve the quality of life of the peoples [7]. From the statistic collected by statcounter, there is 78.46% Malaysians use android smartphones [8]. Hence, the Android Studio software is the best platform to develop a mobile application for the parking system [9]. Lastly, Internet of Things (IoT), a system of interrelated computing devices, mechanical and electrical devices that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction [2].

Due to the motivations that are mentioned in the above, an effective parking system with high performance and the low installation cost which is suitable to open space parking area is needed to be developed to improve the current parking system in Malaysia. The proposed parking system applies the concept of Internet of Things (IoT).

1.4 Problem Statement

Although there are various parking system developed in Malaysia, the development of on-street parking system is always be neglected due to limited resources. Traffic congestion and double-parked issues are always occurred at the crowded area especially at office area or at food court area. This is because the current on-street parking does not install any sensing devices to detect the status of vacancy of parking spaces. In Malaysia, there are no security camera installed on the street. Without the vision based sensor, the availability of the on-street car parks is unknown. No one will know the car parks' availability status at the desired location unless the drivers reach there. This leads the drivers spending more time in searching the parking space lot by lot.

Another problem that is encountered by on-street parking system in Malaysia is that there are no any platform provided to the users to get real-time information of the availability of car parks. Government does not provide any web application or mobile application to the users for checking the parking availability. Most of the parking vacancy display are just showing the number of the available parking but not the exact position of the car park. Furthermore, although there are some parking payment applications provided by state government such as Malacca and Kuala Lumpur, those payment systems can only to be used in town area. For most of the areas such as Puchong, Subang and Ayeh Keroh, the parking payment system still requires the purchase of coupon. The coupon system is not a convenient system for drivers and motorists, especially for those who seldom visit the area. Besides, it is difficult to purchase the coupons since there are very less coupon sellers. There is a possibility that the driver may get fine at the moment he or she are searching for parking coupon.

To improve the current on-street parking system, an application should be developed to deliver the real-time information about the status of availability of car parks at the desired location. In addition, sensors should be installed at the on-street parking area to monitoring the car parks. Lastly, a reliable detection system should be developed to identify the availability of parking so that the users can obtain the most accurate information.

1.5 Objectives of Project

This project embarks to achieve the following objectives upon its completion:-

i. To design and develop an on street parking system with the integration of mobile application for providing real-time information to the users.
ii. To evaluate the performance of both contour parking vacancy detection system and Haar Cascade parking vacancy detection system in terms of accuracy in detecting the status of availability of car parks.

1.6 Scope of Project

In accomplishing the intended goals of the project, several limitations were defined as follows:

i. The parking system is focused on on-street parking area. The prototype is built with three different types of parking which are parallel parking, angle parking and perpendicular parking. Each type of parking has 4 parking lots.
ii. The size of the parking lots are designed for the toy cars. The size of the parking lots at the prototype hardware is smaller than the size of actual parking lots.
iii. A yellow circle with number label is placed at the center of the parking lot. When an IP camera detects the number along with its yellow background, it shows the parking lot is available.

iv. The evaluation of the system is done at daytime. The experiment setup is carried out at the bright condition which has sufficient lighting condition.

v. Huawei smartphone is used as an IP camera for monitoring the parking area.

vi. Camera tripod is embedded with the camera to provide stability and prevent the camera from capturing an obstructed image.

vii. A mobile application is developed using Android Studio. The mobile application can be used by android users with the internet connection.

viii. Intel Up Squared board is used as a server and a processor for computer vision algorithms. The Ubuntu OS and OpenCV software are installed in it.

ix. OpenCV python is used as a medium of processing the videos or images that are captured by the IP camera.

x. Firebase is used as a cloud system to store data obtained from OpenCV and transfer the data to the mobile applications.

1.7 Thesis Outline

This research project is documented and structured as follows. Chapter 2 provides the literatures review based on the previous findings and researches. Chapter 3 highlights the methods used in this project in order to achieve the objectives of the research. The analysis and discussion on the results of the experiments are stated in Chapter 4. Lastly, the conclusion and recommendations about the research is emphasized in Chapter 5. All the references and appendices related to the research is attached at the end of the report.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, literature review related to the on-street parking system are done by referring the trusted resources such as articles and journals. First of all, the concepts and requirements of an effective parking management system. Next, the background of various existing parking system are described. Reviews on related works are studied and compared. The evaluation and comparison are done based on limitations of the system, method of detection and method of analysis.

2.2 Effective Parking Management System

Generally, an effective parking management system needs a server or a gateway that are used to be a bridge to interface hardware (sensors) and software (cloud, mobile application and website). Once a car enters or leaves the parking area, the server will send the real-time information to the database via the internet. When a car parks over the sensor, the information will be transmitted wirelessly to the gateway and the occupancy will be reported instantly to the users via application. When the car leaves the parking lot, the application will be automatically connected to the payment method system. Hence, it is more convenient for the users to get the real-time information of the parking area and also save their time to search the payment machine. For the authority, he can identify non-paying cars with the use of mobile application. This method makes the parking workers can work effectively. Figure 2.1 shows the basic concept of an effective parking management system.

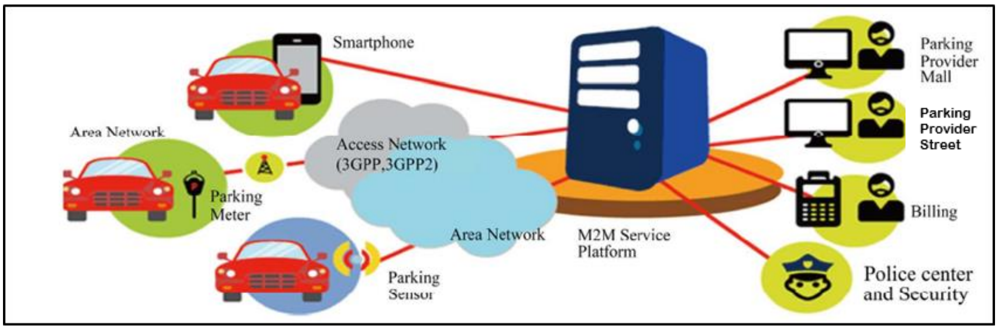


Figure 2.1: The Basic Concept of An Effective Parking Managemnet System

An effective parking management system should consists of the following specifications [10]:

a) Able to optimize the usage of parking spaces.
b) Able to minimize the traffic congestion in the city.
c) Able to guide the users or drivers to the available parking lot.
d) Able to detect the vehicle occupancy in real-time accurately.
e) Able to provide the real-time information of parking availability to the users.
f) Able to provide a media for the users to pay parking fee.

2.3 Comparison of Existing Open Space Parking System

Table 2.1: The Comparison Table Between The Existing Open Space Parking Systems

Articles	Sensing Method	Sensors/Spots	Accuracy	Limitations
[11],[12]	Mobile Sensing Unit	Not Necessary	>= 90%	The performance of the system relies the accuracy of GPS information, sensing unit need to be mounted in a car.
[13]	Camera	Not Necessary	> 96%	The performance of the system is affected by the shadows produced on the parking lots, the solar reflection from the vehicles.
[14],[15]	Magnetic Sensor Node	1	> 98%	Sensor nodes have limited computing power and memory

[16]	Ultrasonic Sensor	1	No mentioned	Require wiring works to install ultrasonic sensor.
[17]	GPS/ accelerometer sensors	Not Necessary	> 97%	No external sensors such as camera, wireless sensor and ultrasonic sensor, everyone must have smartphone with bluetooth sensor.
[18]	Video camera	Not Necessary	No mentioned	Require large amount of storage memory.
[19]	Ultrasonic sensor	2	>= 90%	Require more than 1 ultrasonic sensor to reach high accuracy of detection of parking availability.
[20]	Photoelectric Sensor	2 at the speed bump	> 98%	The maximum speed allowed at the parking area is 20km/hr and the information provided is for whole parking area but not specific for a parking lot.
[21],[22]	Web camera /Camera	Not Necessary	> 90%	Images will have shadow when there are no sufficient lighting condition.

In this research, the discussions and justification of existing parking system are focused on the on-street parking area. There are various methods of detection used in existing parking system to identify the status of the parking availability. Vladimir Coric and Marco Gruteser [12] proposed that the preinstalled parking sensors that are mounted on the passenger side of a vehicle are able to detect the presence of parked vehicles on the street by measuring the distance from the nearest obstacles beside the street. The mobile sensing unit helps to collect data about the status of parking availability. In addition, in this system, the mobile sensing unit is not required to be implemented at each of the parking lot. The experiments that are done in article [11, 12] proved that the performance of the parking system that uses mobile sensing unit hits the accuracy more than 90%. The accuracy of the parking system is affected by the stability of the GPS. However, this sensing method is not recommended to be used in Malaysia because most of the Malaysians do not want to spend money on installing the mobile sensing unit in their own cars.

The other sensing method that are used in existing parking system is ultrasonic sensor. Ultrasonic sensor emits the sound wave of specific frequency to detect the presence of a vehicle. The ultrasonic sensor needs to be installed in each of the parking lot. In [19], it mentioned that each parking lot requires 2 units of ultrasonic sensors to increase the accuracy of detection of parking availability. The accuracy of the system that are proposed by Dr.V.Kepuska and Humaid Alshamsi [19] reaches the accuracy above 90%. However, the use of ultrasonic sensor as the sensing tool to be used to monitor the status of open space parking availability is not suitable. This is because the ultrasonic sensor is impossible to place on the ground of open space parking area. The ultrasonic sensors will be damaged or broken when the car runs over the sensors and crush them. In addition, the implementation of ultrasonic sensors to the open space parking area needs the wiring works which lead to the increasing of cost. The accuracy of the parking availability detection using ultrasonic sensors is affected by the environmental factors such as the surrounding temperature, the types of the surfaces used and so on. Due to the cost and unsuitability, the ultrasonic sensor is not be recommended to be chosen as the sensing method.

The magnetic sensor node is also be used to be installed in parking lots to detect the status of the vacancy of parking spaces. Zusheng Zhang [14] proposed a on street parking system based on wireless sensor networks (WSNs). To evaluate the performance of this system, 82 of magnetic sensor nodes are installed on the open space parking area and the system is ran for one year. This system requires few components which are base station, routers, server and sensor nodes. The evaluation of the accuracy of the system is done with two types of algorithm which is the algorithm that is developed by Zuzheng Zhang and his team and Adaptive Threshold Detection Algorithm (ATDA). After the experiments, it proved that the algorithm that are developed by them has a higher accuracy than the ATDA which is more than 98%. The problem encountered by the on-street parking is the strong noise disturbance. To solve this problem, Hongmei zhu and Fengqi Yu [15] use the normalized cross-correlation (NCC) method. By using this method, the accuracy of detection can reach an accuracy of 99.33% for arrival and 99.63% for departure. However, there is a tradeoff between sensitivity of magnetic signal that will result in the detection of vehicles in adjacent parking place. Due to the weakness of the magnetic sensor, the magnetic sensor is not suitable to be implemented in on street parking area in Malaysia because the noise of surrounding environment such as construction works, motorcycle noise and other source of noise.

In [17], a system that is named PhonePark was developed by Bo Xu and his team members. They proposed to use the GPS or the accelerometer sensor that is built-in in the drivers' mobile phone to detect the locations that the drivers parked their car. In this system, there are no extra external sensors such as camera, ultrasonic sensor and photoelectric sensor to detect the parking availability. Besides, to estimate the parking availability in real-time, they compute the historical parking availability profile for an arbitrary street block using an algorithm. The accuracy of detection of parking availability reached more than 98%. However, this method is not a good solution because not all the drivers carry a mobile phone and they may not install the PhonePark system in their phone. In addition, false information may be produced due to the GPS errors, transportation mode detection error and Bluetooth pairing errors. Therefore, this method is not suitable to be used.

Photoelectric sensors are also be used to deploy on the access roads into and out of the parking area. In [20], the wireless sensor node (WSNs) concept is used in this system. The sensor node is connected to two sensors to detect the passage of vehicles. The two photoelectric sensors was place in a speed bump to monitor the exit and entry process at the parking area. The data obtained will be sent to the data centre. This proposed system is tested at the Tafira Campus of University of Las Palmas de Gran Canaria. The result shows that the accuracy of this system in detecting the passage of vehicles is quite high, which reaches more than 98%. However, this system has some limitations such as this system can only detect the number of cars in and out the parking area but it cannot give the specific parking lot that are available to be parked. Moreover, the speed limit that the car can cross over is 20km/h only. This means that if the car's speed is more than 20km/h, the accuracy may be decreased. This method is not suitable to be implemented in Malaysia because the bump speed in Malaysia is different with other Europe country and it is very difficult to install photoelectric sensor inside the bump speed.

Next, the camera is also used to capture the images at the parking lots. The camera is installed at a fixed position such as at the street lamp to monitor the parking area. Articles [13, 22, 23] support that the vison based system is the best solution for monitoring and identifying the parking availability at the on-street parking area. There are two analysis method that are provided to analyse the captured images from camera which are Convolutional Neutral Networks (CNNs) which is proposed in [13] and image processing techniques [21, 22]. The vision based system is divided into two categories which are video camera and web came or CCTVs camera. Vision based system provides the accuracy of more than 90%. The accuracy of the detecting the status of vacancy of parking lots is depend on the quality of the camera such as the resolution, pixels and the cost of the camera. Video camera is not recommended to be used in the parking system because it requires large amount of memory to store the video of the parking area. However, the images that are captured by the camera have shadow when it is captured under the insufficient lighting condition. The sharpness and quality of the images can be improved by using image processing techniques such as image enhancement, image segmentation and mask processing filter.

In conclusion, the vision based system is selected as the sensing method to identify the parking availability. This is because camera as sensor can provide the real-time information and exact position of parking lots to the drivers. In addition, the camera can improve the security of the on-street parking area such as motion detection, face recognition, car plate verification and so on. In addition, due to the limited space at the open space parking area, the camera is the most suitable sensor to be implemented. The camera can be installed at the top of street lamp. Although the cost of a camera is higher than ultrasonic sensors, magnetic sensors and photoelectric sensors, it can be used to detect more than one parking lots. In [21], the article stated that a camera is implemented in India to identify the status of the vacancy of 10 parking lots. Moreover, the installation of camera at the on-street parking area is not require any cost on wiring works. Therefore, the vision based system is a cost-effective method to develop an effective on-street parking system. The existing on-street parking systems are focused on monitoring the parking area only. However, the characteristic of an effective on-street parking system should provide the convenient to the drivers such as check the parking availability, reserve the parking lot, online-payment and so on. Hence, in this project, an effective on-street parking system with low cost and high accuracy should be developed.

2.4 Comparison of the Computer Vision Algorithm Used for Detecting Car Parks

Table 2.2: The Comparison Table of Computer Vision Algorithms Used for Parking Detection

Article	Algorithm Used	Accuracy	Weakness
[24],[25]	Haar Cascade Classifier	90%-100%	• The camera used must be fixed at a fixed location. • Detect extra objects (other than cars)
[26]	Contour tracing, edge detection and segmentation	Not mentioned	• Complex image processing algorithm • Need more time to process the images
[27]	False contour detection	99%-100%	• The efficiency will go down when car and parking area are of the same colour.
[28]	AdaBoost Classifier	Not mentioned	• Different object should be trained seperately
[29]	Color consistency, Shadow detection and background generation	Not mentioned	• The performance of the detection is affected by the angle of elevation from vehicle to the camera.

The computer vision algorithms are very important for vision-based on-street parking system. This is because each algorithm has its strength and weakness in detecting an object. Some detection method needs to work with others to get a better performances. In [26], the captured image is first undergoing pre-processing step to reduce background noise and also enhance the contrast of the image. Next, it proceeds to contour tracing, edge detection and segmentation parts. The combination of these image processing technique has succeed to recognize the vehicle. If each of those image processing technique works alone, the vehicle detection system will not get the most optimize results. Another example given by the article [29], it shows that the combination of three techniques in vision-based parking system has improved the efficiency of detection of parking vacancy. However, the combination of image processing techniques makes the system become complicated and causes the processing time increasing.

From the findings shown in Table 2.2, it depicts that the Haar Cascade Classifier is one of the popular computer vision algorithms to be used to train a specific detection. Both articles [24] and [25] explained that the parking detection system with Haar Cascade Classifier has achieved the accuracy of 90% to 100% in detecting the parking vacancy. Haar Cascade Classifier uses the changes in contrast values between adjacent groups of pixels rather than actual pixel values to determine common Haar features within an image. Haar-like features is available for the purpose of determining the information range from edge features, line features and center-surround features. The weakness of this classifier is that the performance of the detection system will be affected by environment artifacts [25]. For example, light intensity variation is a critical factor in detection. If the condition of captured images is slightly different with the trained condition, the performance is dropped. However, the accuracy of the Haar Cascade Classifier can be improved when large set of positive images and negative images are added on the training of classifier.

Another common computer vision algorithm is contour-detection method. In [30], the contour approach can be divided into two which are active and false detection. Active contour is a vibrational approach to contour localization. Besides, active contour is an energy minimizing approach guided by the external forces and influenced by the image forces that attract the initial drawn curve to the contours. Furthermore, false contouring is known as "pasteurizing" or "quantization noise," it refers to the creation of false edges or outlines where the original scene had none. False contouring happens because of reduction in gray levels of an image Article [27] shows that the parking detection system with false contour detection method has hit the accuracy of 99% to 100% by comparing with the Block Based Classification, Edge Detection, Twin ROI, and Inverse Binary Segmentation. However, the efficiency of the detection system with false contour detection will go down when car and parking area are of the same colors. For the AdaBoost Classifier that is mentioned in [28], the training method is almost same with the Haar Cascade Classifier. However, the discussion from the article [31] shows that the Haae Cascade Classifier has a better performance than the Gentle AdaBoost.

From the analyzation and comparison of the articles, Haar Cascade Classifier and contour-detection method are chosen to be used in this project. This is because the performance of both computer vision algorithms are high and stable. In addition, both methods are very useful in detecting the availability of car parks.

2.5 Summary

Various articles and findings about the existing on-street parking systems are studied and analyzed to find out the optimized solution to develop an on-street parking system with low cost and high performance. After the justification and comparison, the vision based system using a camera is chosen as the sensing method to detect the parking availability. After analyzation of the computer vision methods used by the existed vision-based parking system, Haar Cascade Classifier and contour detection method are selected to be used to evaluate the accuracy of the on-street parking vacancy detection system. Moreover, an application is designed and developed to provide the real-time information to the drivers so that the drivers can save times and cost in searching parking lots. Notification system is also be considered in this project to remind the drivers about the latest updated information. The methodology of developing this on-street parking notification system will be discussed in details in next chapter.

CHAPTER 3

METHODOLOGY

3.1 Introduction

A clear explanation on the design of microprocessor-based on-street parking notification system is described in this chapter. The concerns of this project are developing an on-street parking system by implementing the IoT concepts and also design a parking vacancy detection system using computer vision algorithm. This microprocessor-based on-street parking notification system is divided into 3 parts which are hardware, software and interfacing. The equipment used in this project is listed with its specifications. In this chapter, the development of mobile application and detection system is also explained in details. The interfacing between hardware and software using IoT ecosystem is also highlighted. After that, several experiments to evaluate the performance of the data synchronization between OpenCV, Firebase and mobile application and the accuracy of parking vacancy detection system at different conditions such as different types of car park and different environments. The experimental setup and parameters used are also presented clearly.

3.2 System Overview

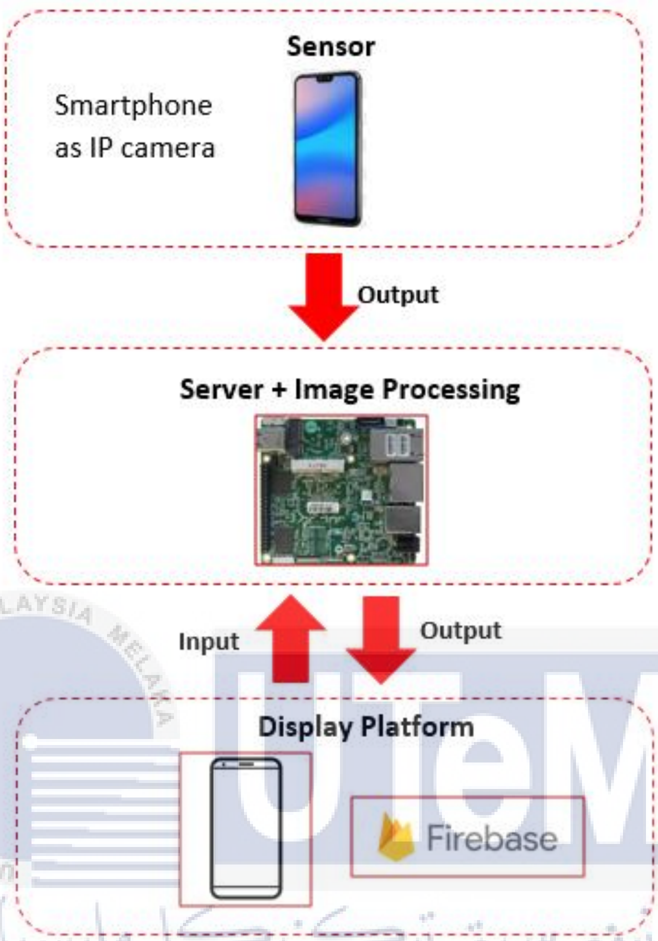


Figure 3.1: System Overview of This Project

Table 3.1 shows the system overview of microprocessor-based on-street parking notification system. This system are divided into three main parts which are sensing device, processing unit and display platform. Huawei smartphone model P20 is chosen as the sensing device for this system. This is because the resolution of the camera is high. In addition, the images and videos captured can be directly transferred for image processing purpose. Firstly, IP Webcam application is downloaded from the PlayStore. Then, the IP address is obtained and ready to be used in this system.

Intel UP Squared board is selected as a server and a processor. It acts as a medium to interface software and hardware. Besides, Intel UP Squared board is also used as a processing unit to process the images and videos that are obtained from IP camera. To implement computer vision in this system, Ubuntu OS and OpenCV software are installed in Intel UP Squared Board.

For this project, Firebase cloud system and Android mobile application are used to display the results from the parking vacancy detection system. The mobile application must be linked with Firebase cloud system for data synchronizing. The results obtained will be directly stored to Firebase cloud system after processing the captured videos or images using computer vision algorithm. Then, the status of availability of parking vacancy is obtained from the Firebase cloud system and displayed on the mobile application. The users can grasp the latest real-time parking vacancies' status via mobile application. The equipment used in this project are listed in Table 3.1. All the specifications of each component is also shown.

Table 3.1: The Selected Components with Their Specifications

Components	Specifications
IP camera	**Model** Huawei P20 **Wi-Fi** Wi-Fi 802.11 a/b/g/n/ac, dual-band, WiFi Direct, hotspot **Camera** ▪ Primary Dual: 12 MP + 20 MP , Leica optics, 2x lossless zoom, AIS, phase detection and laser autofocus, dual-LED dual-tone flash ▪ Secondary 24 MP, autofocus, f/2.0, 1080p **Video** 2160p@30fps, 1080p, 720p@960fps
Intel UP Squared Board	**Processor** Intel® Celeron™ N3350 (up to 2.4 GHz) Intel® Pentium™ N4200 (up to 2.5 GHz) Intel® Atom ™ E3940 (up to 1.8GHz) **Graphics** Intel® Gen 9 HD, supporting 4K Codec Decode and Encode for HEVC4, H.264, VP8 **Ethernet** 2x Gb Ethernet (full speed) RJ-45 **OS Support** Microsoft Windows 10 (full), Windows IOT Core, Linux (ubilinux, Ubuntu, Yocto), Android Nougat

3.3 Development of Microprocessor-based On-street Parking System

To develop an effective on-street parking system, the designs of detection system, mobile application and integration system are very important. In this section, the design of each part of the parking system is described in details with flowcharts and methods used. In subsection 3.3.1, the design of parking vacancy detection system using computer vision algorithms that are mentioned in Chapter 2. Next, the methods of designing a mobile application with Android Studio are depicted. Lastly, the approach on the integration between hardware and software using IoT concepts is discussed.

3.3.1 Design of Parking Vacancy Detection System

To design parking vacancy detection system, two common computer vision algorithms are used. There are contour-detection method and Haar Cascade Classifier. Contour-detection method is a simple algorithm to detect the yellow circle in the center of car park by applying the concept of joining all the continuous points which have same color or intensity. The parking vacancy detection system with contour-detection method is designed as descried in below:

i. All the possible areas which have continuous points along boundary with same intensity are recognized.

ii. Those detected areas are marked by red circles with labels of available.

iii. To increase the accuracy of the contour-detection method, the condition of radius range of the detected areas is added to the python code.

iv. Those detected areas which the radii are not in the range of 30-40 pixels are eliminated automatically.

v. The remaining detected areas are verified by the system to make sure that they are the yellow circles in the center of car parks. If they are the yellow circles in the center of car parks then their coordinates are shown else the error found by the system.

vi. Their coordinates is used to determine the position of each car park.

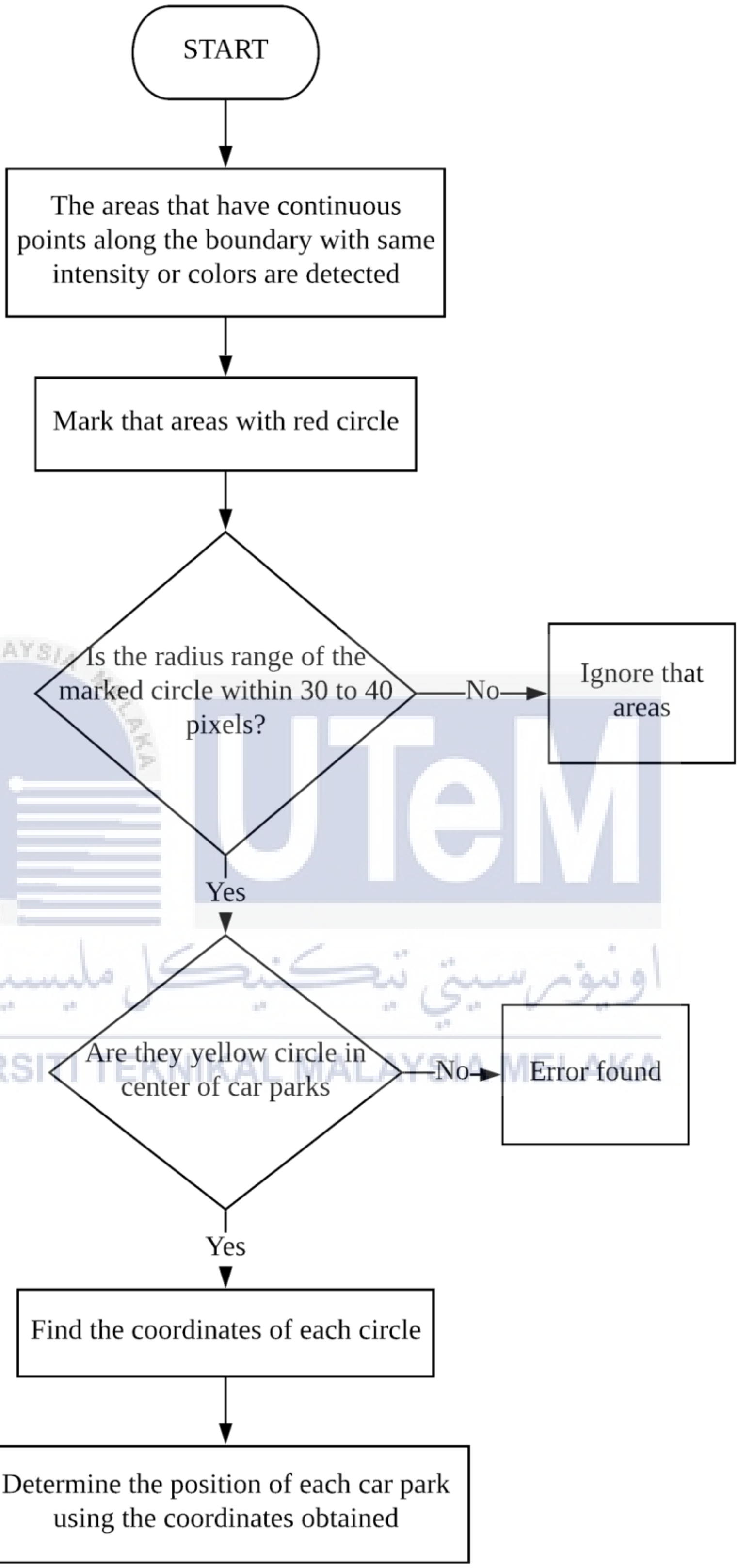


Figure 3.2 : Flowchart of The Parking Vacancy Detection System with Contour-detection Method

The detection system with Haar Classifier is vital for car park vacancy detection. Basically, Haar Cascade Classifier is a classifier which is used to detect the object for which it has been trained for, from the source. It is trained by overlapping the positive image over a set of negative images. Generally, the training is completed on a server and various stages. Better results will be obtained if high quality images are used for the training. In addition, increase the amount of stages for which the classifier is trained is also improve the results obtained. The design of parking vacancy detection system with Haar Cascade algorithm is reported as in below:

i. The cascade trainer GUI is downloaded from website and then it is installed in PC. The cascade trainer GUI is shown in Figure 3.3.

Figure 3.3 : Cascade Trainer GUI

ii. A folder is created. Positive and negative folders are necessary to be added in the created folder for training purpose.

iii. The image used for training is converted to grayscale image. Then, the grayscale image is cut into 200 parts using IMGonline.com. The 200 small images is categorized into positive image and negative image. The obtained positive images is deleted and the obtained negative images are stored at the negative folder. Next, the grayscale image is used to crop the positive images that used in training. All the cropped positive images are saved in positive folder.

iv. The path of sample folder is added into cascade trainer GUI to start the training. Then, the negative image count is changed to 300. The training is started once the start button is clicked.

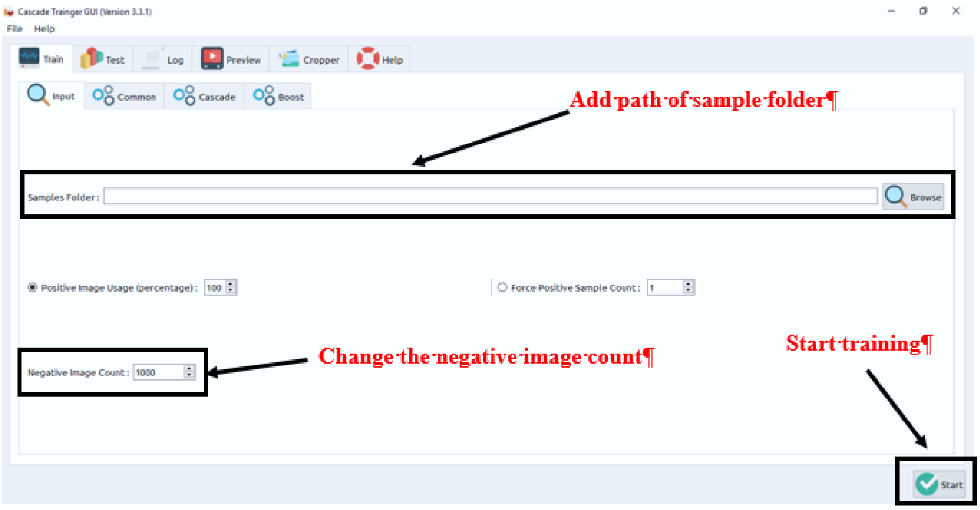


Figure 3.4 : Add Path of Sample Folder To Cascade Trainer GUI

v. The .xml file is created by cascade trainer GUI after completing the training. The .xml file is used to add in program code to recognize the parking space.

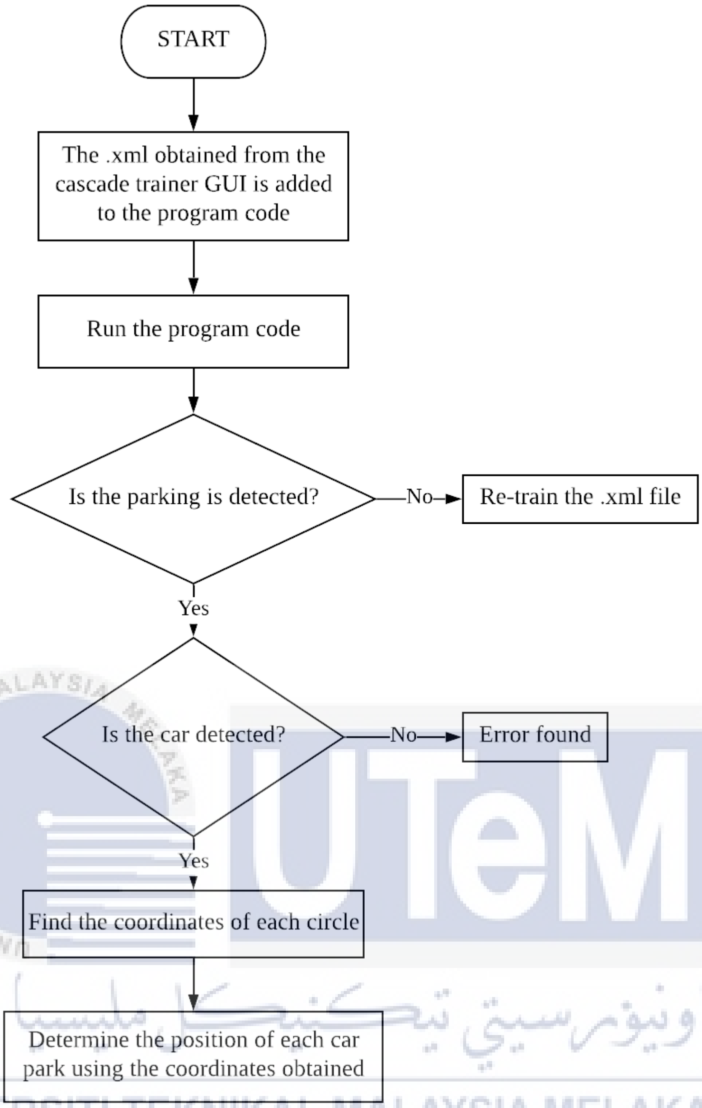


Figure 3.5: The flows of Developing Haar Cascade Algorithm

3.3.2 Design of Mobile Application

In this project, Android Studio software is used to develop a mobile application. The mobile application architecture in this project consists of four main parts which are end user, system admin, parking system controller and mobile application. The end users interact with the mobile application such as check the status of the vacancy of parking lots and login or register into the system. The system admin controls the system and make sure that the data are well-maintained and updated. Besides, the parking system controller plays the role of ensuring the IP camera is working fine, the accuracy of the data and so on. For the mobile application, a graphical user interface (UI) Android based mobile application provides the following functions which are sign in, log out, reset password, parking status checking and pop-up notification. The mobile application connects to the s once the user logs in to his or her account and then the real-time information is received by the user. The design of mobile application is presented in Figure 3.6.

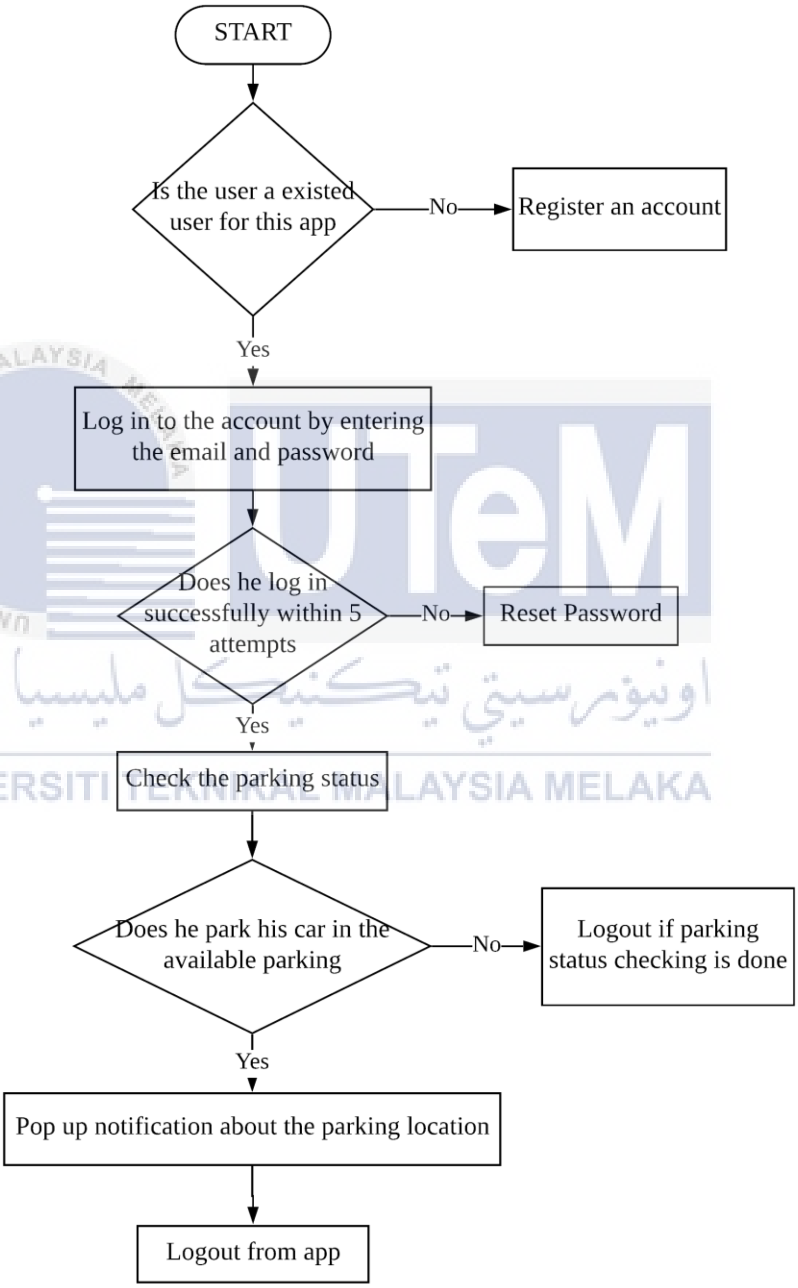


Figure 3.6: Flowchart of Mobile Application

First, the driver is required to download the mobile application from the Google Play Store in the Android smartphone. After downloading the application, the driver needs to enter his or her personal details such as email, mobile number and password to register as a user for this application. The user needs to log in to his or her account by using the email and password. If the user cannot log in successfully within 5 times of attempts, the account cannot be logged in anymore unless the password is reset. Once he or she resets the password, an email is sent for password resetting. All the user interfaces (UI) for registration, login activity and password resetting are shown in Figure 3.7.

After logging successfully, the status of car parks can be checked by the user. The user is able to get the latest availability of parking vacancy at the desired location at his fingertips at anywhere and anytime. Once the user parks into the available parking, the pop-up notification is jumped out to show the current position his vehicle. If the user does not want to check the parking status, he can leave the app by pressing the logout button. The UI of the drawer of parking application is designed as shown in Figure 3.8.

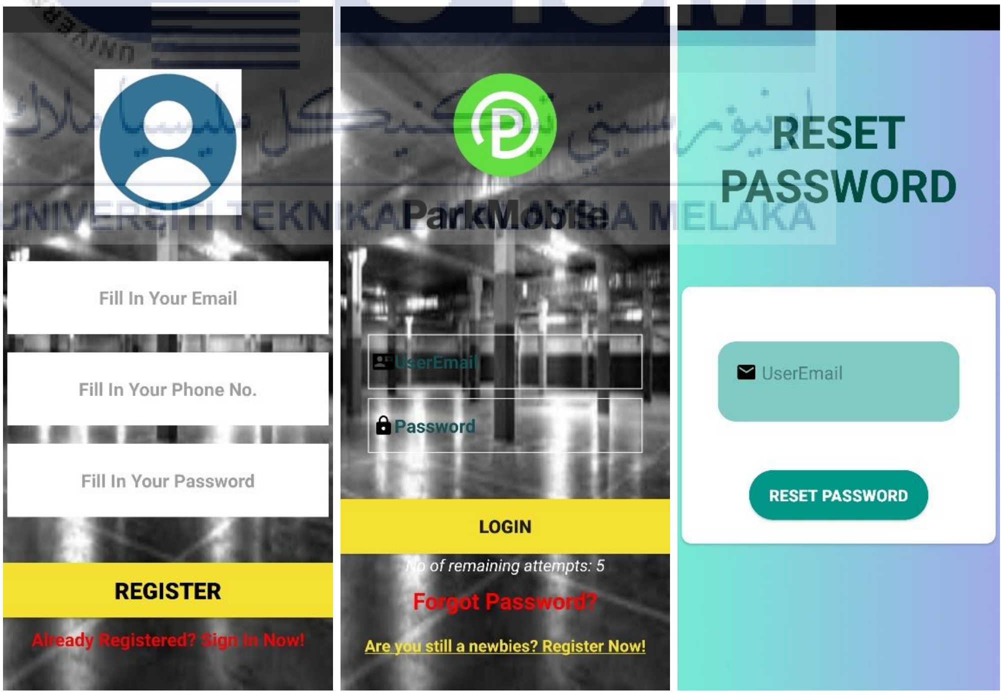


Figure 3.7: User Interfaces(UI) of Registration, Password Reset and Login Activity

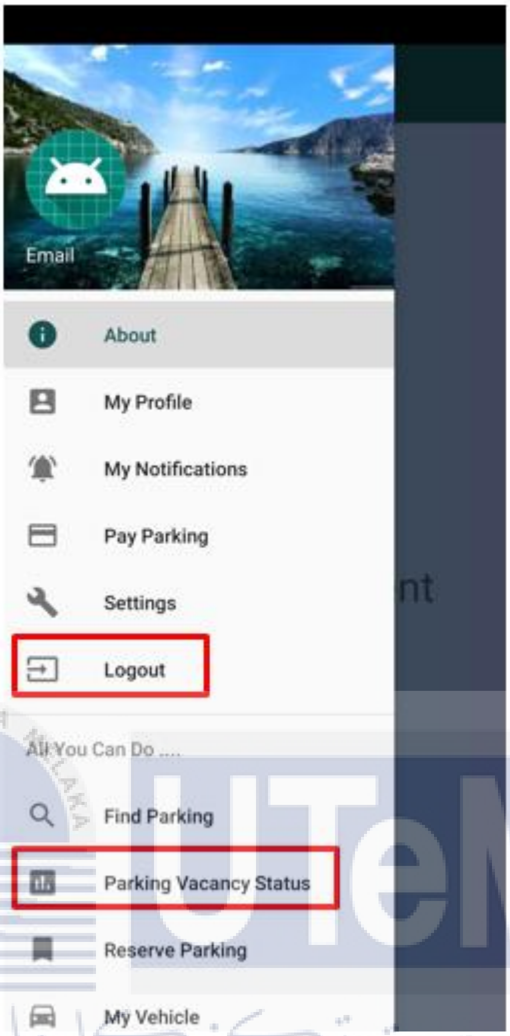


Figure 3.8 : User Interface of The Drawer of Parking Application

3.3.3 Design of the Data Synchronization System Using IoT Ecosystem

Data Synchronization System is a major concern of on-street parking system to transferring the data. In this project, the information from the captured images or videos is necessary to be transferred to the users so that the users are able to know the latest real-time information from time to time. The integration between OpenCV, Firebase cloud system and Android mobile application. The design of the data synchronization system using IoT ecosystem is described clearly as below:

i. First, the Firebase account is registered using gmail and a project is created. Then, the URL shown in the real-time database is copied and pasted to the program code. The steps are demonstrated in Figure 3.9. After completing the steps, the results obtained from the parking vacancy detection system in OpenCV are successfully transferred to Firebase cloud system.

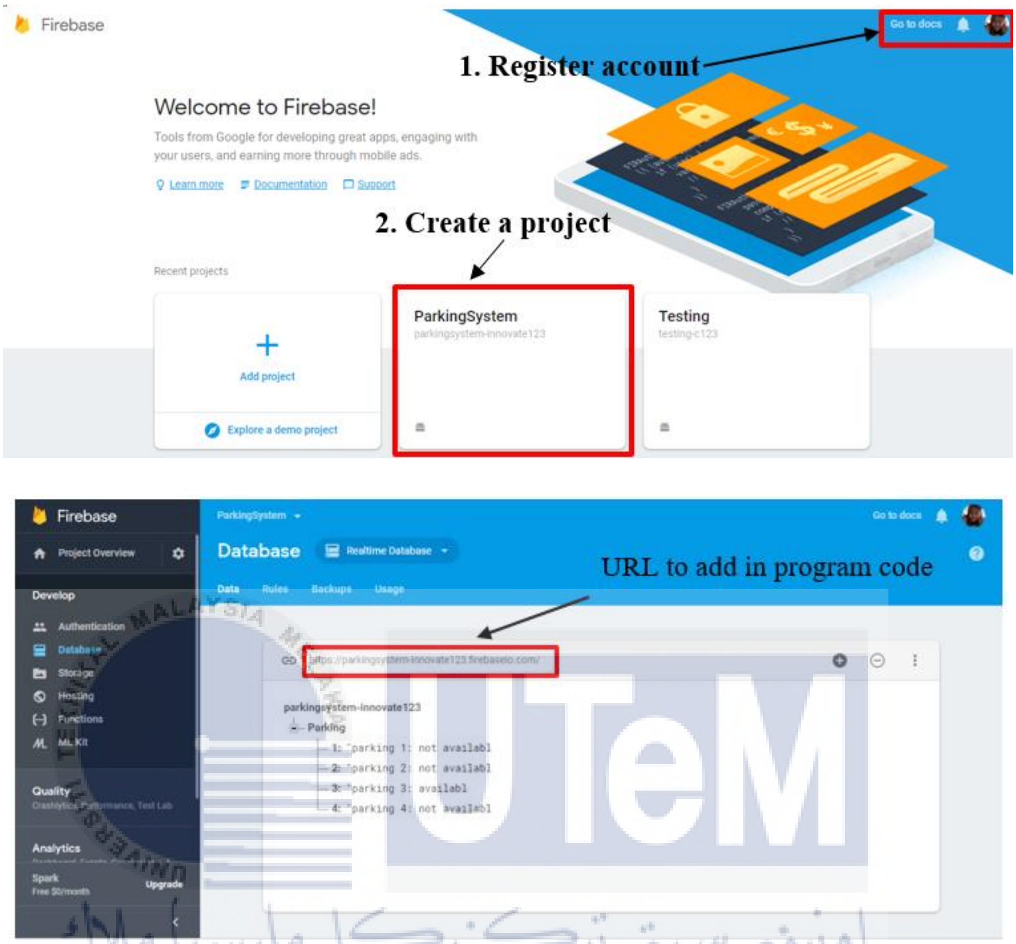


Figure 3.9: Steps to Integrate OpenCV and Firebase

ii. Then, the mobile application is linked with Firebase cloud system. The authentication and real-time database is necessary to linked between Firebase and Android Studio by inserting the required dependencies. Then, the SHA-1 in signing report is copied and pasted to Firebase. All the details is shown in Figure 3.10.

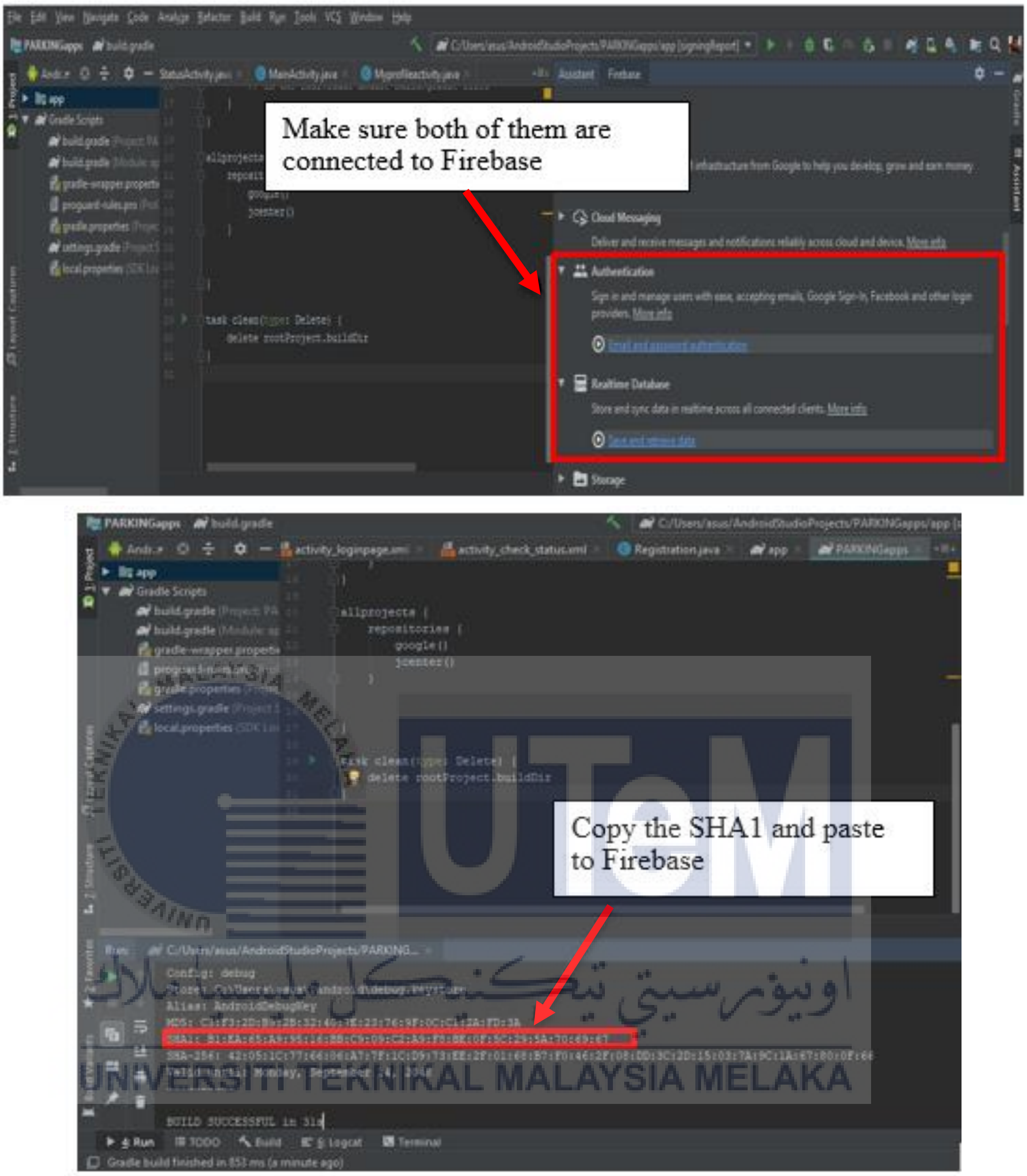


Figure 3.10: Linking Approach Between Firebase and Android Application

After completing all the designs of each system, this project is successfully built and developed. The status of parking vacancy can be updated from time to time by detection system and the data obtained is transferred to end users by data synchronization system. The performances of the data synchronization and parking vacancy detection are evaluated by conducting the experiments that are mentioned in section 3.4.

3.4 Experiment Design

Few experiments are carried out in this project to evaluate the performance of the parking system in terms of accuracy. In this project, 3 different types of parking were used. For each type of parking, it consisted of 4 car parks. To develop an effective on-street parking system, the efficiency of a parking system to provide an accurate real-time data is very important. Hence, in this section, the evaluation on the performance of data synchronization system and parking vacancy detection system are done. The experiments are repeated with different cases to get a reliable evidence. Then, the experiments for each case is repeated for few times to calculate the successive rate. The successive rate is calculated is using the formula shown in equation 3.1. All the approaches and parameters used in the experiments are also mentioned in this section.

$$success\ rate, R\ (\%) = \frac{number\ of\ success\ sample}{number\ of\ sample\ used} \times 100\% \qquad (3.1)$$

3.4.1 Evaluation on the Performance of Data Synchronization System

The evaluation on the performance of data synchronization system was done using 10 different cases. The cases with different number of cars in the car parks at 3 different types of parking were chosen to be tested. The results displayed at Firebase cloud system and Android mobile application were observed and recorded as shown in Figure 3.11. The observations obtained were compared with the reference images to justify the successive rate of data synchronization system. All the results and observations were tabulated in a table.

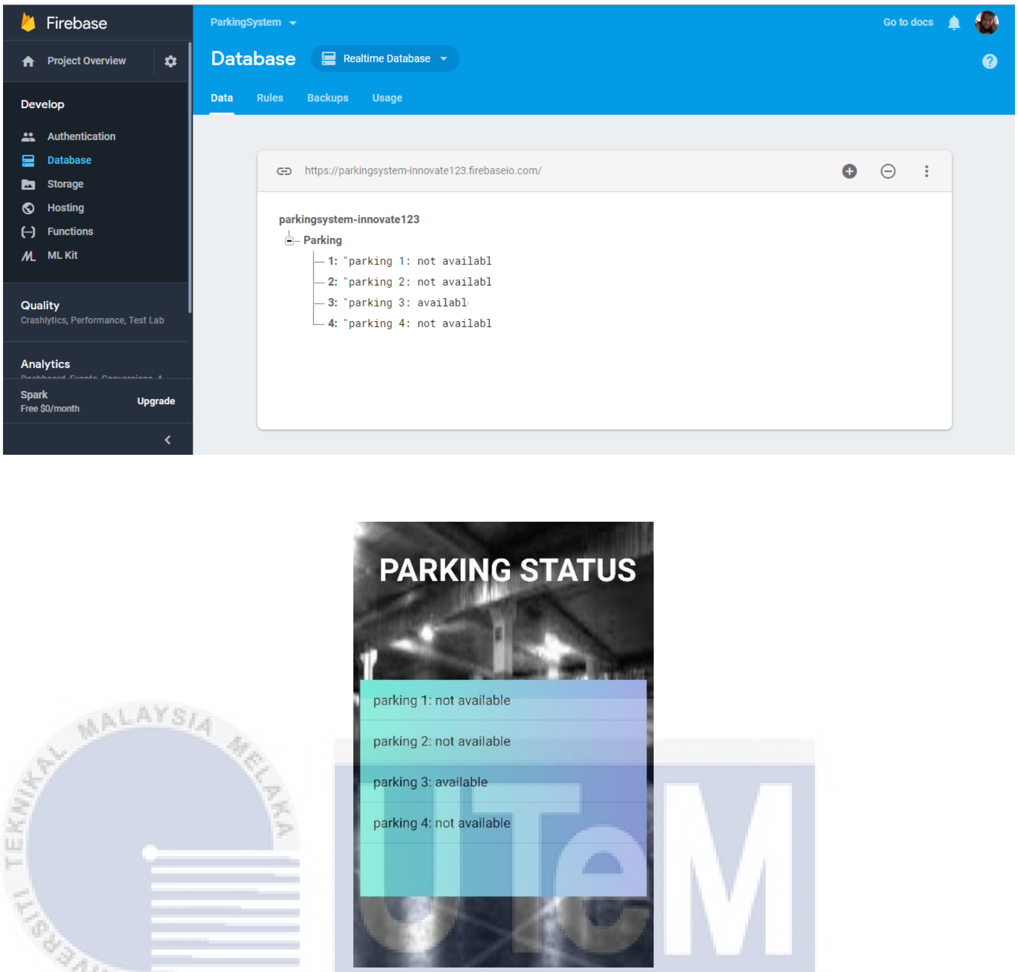


Figure 3.11: The Observations from The Display Platforms

3.4.2 Evaluation on the Performance of Parking Vacancy Detection System

To evaluate the performance of parking vacancy detection system, 3 experiments were designed. The assessment on the accuracy of the contour-detection method in detecting the parking vacancy was done with six differents of radius range. The six radius ranges used in the experiment were no radius range settings, 10 to 40 pixels radius range, 15 to 40 pixels radius range, 20 to 40 pixels radius range, 25 to 40 pixels radius range and 30 to 40 pixels radius range. First at all, the image of the reference image was added in the program. Secondly, the program code was run for the evaluation. Thirdly, the radius range was changed to another value as shown in Figure 3.12 for evaluating the accuracy of contour-detection system. The experiment was repeated to determine the most preferred radius range that used for next experiment. The experiment setup for software and hardware was shown in Figure 3.13.

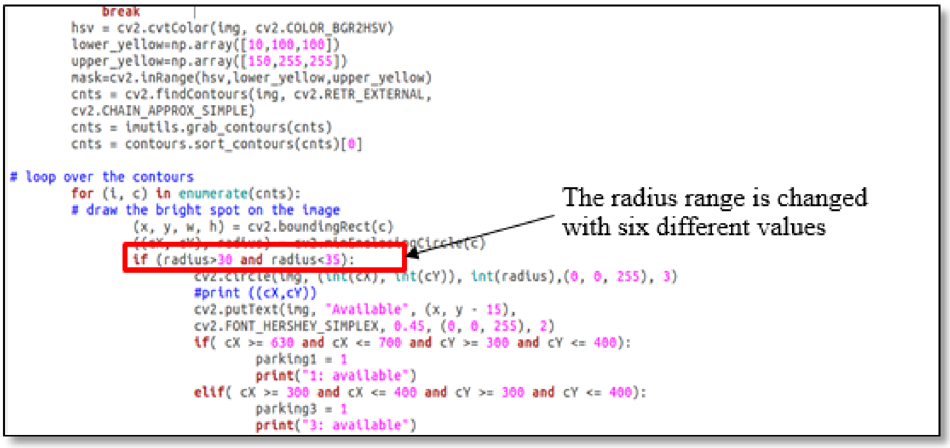


Figure 3.12: The Selection for The Most Preferred Radius Range

Figure 3.13: Experiment Setup for Software and Hardware

After the preferred radius range of contour-detection method was selected, the comparison on the accuracy of parking vacancy detection system with two computer vision algorithms was done by conducting the experiment at three different types of parking. They were perpendicular parking, angle parking and parallel parking. For each type of parking, 5 different cases were used to be justified. The experiment for each case was repeated for 20 times to get the successive rate. The parameters used in the experiment were set as shown in Figure 3.14 to get a reliable result. The heights of the tripod stand was set to 35.0 cm while the distance between the position of tripod and the parking space is 28.7 cm. The experiment setup for each parking type was shown in Figure 3.15, 3.16 and 3.17. The program codes for contour-detection method and Haar Cascade Classifier were shown in Appendix C and D. The experiment was conducted by using the following steps:

1. The image for each case was captured by IP camera and then saved in a folder.
2. The captured images were run in the program code.
3. The status of availability of car parks were shown at the terminal and the proceed images were saved at the created folder.

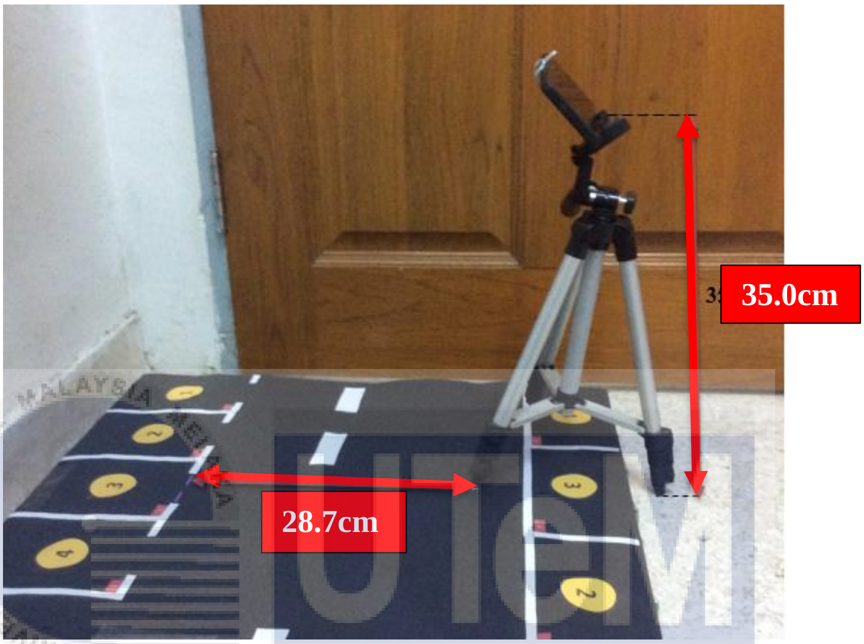


Figure 3.14: Settings of Parameter for the Experiments

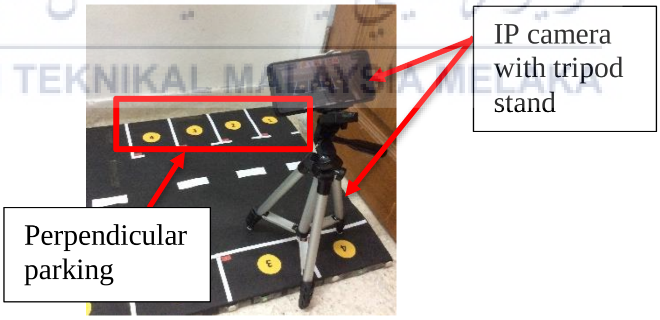


Figure 3.15: Experiment Setup for Perpendicular Parking

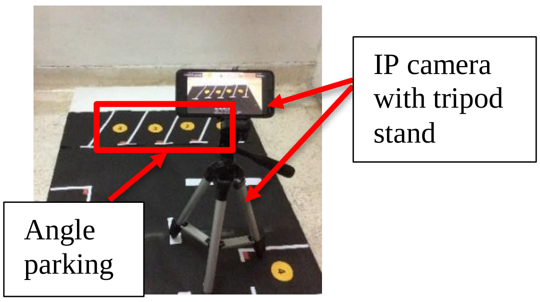


Figure 3.16: Experiment Setup for Angle Parking

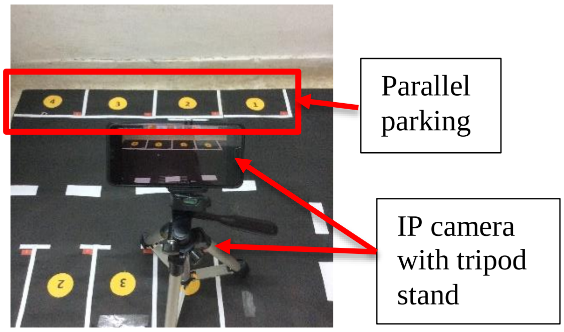


Figure 3.17: Experiment Setup for Parallel Parking

Another experiment for the evaluation of performance of parking vacancy detection system with contour-detection method and Haar Cascade Classifier at different surrounding environments was done. In this experiment, 4 different environments were used to be tested. For each environment, 2 different cases were evaluated. The heights of the tripod stand was set to 35.0 cm while the distance between the position of tripod and the parking space is 28.7 cm for obtaining a more reliable result. The images that shown in Figure 3.18 were used as a reference environment to be trained in cascade trainer GUI. After obtaining the .xml file, 8 captured images from 4 different environments were inserted in the program code for processing. The images obtained were saved in the created folder.

Figure 3.18: The Captured Images From the Reference Environment

3.5 Summary

The overview of the microprocessor on-street parking notification system is explained. The hardware and software used for this system are IP camera, Intel UP Squared board, Firebase cloud system, OpenCV and Android Studio. The design and development of on-street parking system is divided into three parts which are parking vacancy detection system, Android mobile application and data synchronization system. The method of developing these system are explained clearly with the attached flowcharts. Next, the experimental setup and methods are also discussed in this section. These experiments are carried out to achieve the objectives of this project. The findings and observations of the experiment are interpreted and then discussed in next chapter.

CHAPTER 4

RESULTS AND DISCUSSIONS

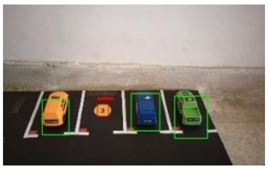
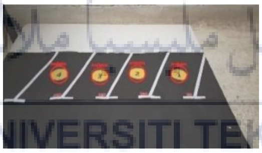
4.1 Introduction

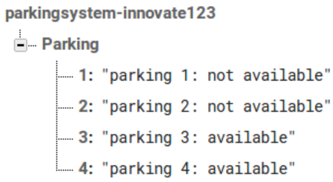
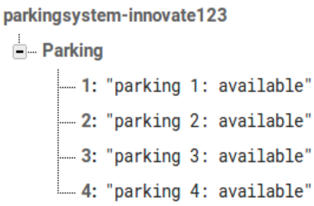
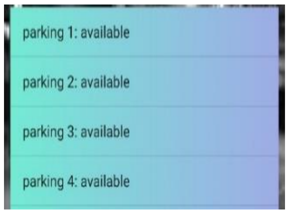
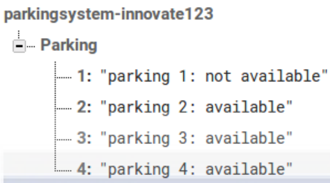
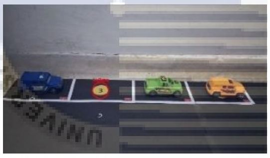
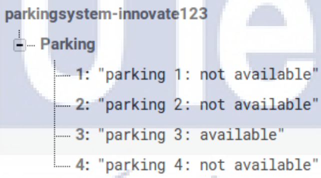
This project mainly focuses on two objectives which are developing an on-street parking system that can provide real- time car park's information to the users and evaluating the performances of the parking vacancies detection system in terms of accuracy. In subchapter 4.2, an experiment on the integration between hardware (IP camera) and software (Firebase and mobile application) using IoT ecosystem is done in order to achieve objective 1. Next, in subchapter 4.3, several experiments are carried out to evaluate the accuracy of the parking vacancy detection system. The computer vision algorithms are used in the experiments are Haar Cascade Classifier and contour-detection method. The interpretations and analyzations of the experimental results are done and discussed.

4.2 Results on the Integrations between Hardware and Software using IoT Ecosystem

The implementation of IoT ecosystem on the on-street parking system is very important to provide real-time information about the availability of car park to the users. Besides, the cloud system is also vital to store the outcomes that are obtained from the OpenCV and then link with an android application to provide real-time data to the users. With the implementation of IoT ecosystem, users are able to know the status of the parking vacancy at their fingertips via the android application. An experiment is carried out by processing the images that are captured by an IP camera using OpenCV. Then, the results are sent to Firebase and the mobile application. This experiment is repeated with 10 different cases to make sure that the information received by the users are always accurate and reliable. Table 4.1 shows the results on the real-time data synchronization between OpenCV, Firebase and the mobile application.

Table 4.1: Real-time Data Synchronization between OpenCV, Firebase and Mobile Application

<table>
<tr><th rowspan="2">Case</th><th rowspan="2">Processed Images</th><th colspan="2">Results</th></tr>
<tr><th>Firebase</th><th>Mobile Application</th></tr>
<tr><td>1</td><td></td><td>parkingsystem-innovate123
Parking
1: "parking 1: not available"
2: "parking 2: not available"
3: "parking 3: available"
4: "parking 4: not available"</td><td>parking 1: not available
parking 2: not available
parking 3: available
parking 4: not available</td></tr>
<tr><td>2</td><td></td><td>parkingsystem-innovate123
Parking
1: "parking 1: available"
2: "parking 2: available"
3: "parking 3: available"
4: "parking 4: available"</td><td>parking 1: available
parking 2: available
parking 3: available
parking 4: available</td></tr>
<tr><td>3</td><td></td><td>parkingsystem-innovate123
Parking
1: "parking 1: not available"
2: "parking 2: not available"
3: "parking 3: available"
4: "parking 4: available"</td><td>parking 1: not available
parking 2: not available
parking 3: available
parking 4: available</td></tr>
<tr><td>4</td><td></td><td>parkingsystem-innovate123
Parking
1: "parking 1: available"
2: "parking 2: available"
3: "parking 3: available"
4: "parking 4: available"</td><td>parking 1: available
parking 2: available
parking 3: available
parking 4: available</td></tr>
<tr><td>5</td><td></td><td>parkingsystem-innovate123
Parking
1: "parking 1: not available"
2: "parking 2: not available"
3: "parking 3: not available"
4: "parking 4: available"</td><td>parking 1: not available
parking 2: not available
parking 3: not available
parking 4: available</td></tr>
<tr><td>6</td><td></td><td>parkingsystem-innovate123
Parking
1: "parking 1: not available"
2: "parking 2: available"
3: "parking 3: available"
4: "parking 4: available"</td><td>parking 1: not available
parking 2: available
parking 3: available
parking 4: available</td></tr>
</table>

7			
8			
9			
10			

From Table 4.1, it shows that the information from the images has been successfully transferred to the mobile application for the users. This experiment is repeated with 3 different types of parking which are perpendicular parking, angle parking and parallel parking. The information displayed on both the mobile application and Firebase cloud are exactly the same with the processed images displayed on OpenCV. Once the situation of the parking area has changed, the data at the parking area is immediately updated to the cloud and the mobile application. Therefore, the results has proved that the IoT ecosystem are implemented perfectly on the on-street parking notification system. In addition, the results have justified that the on-street parking notification system is able to provide an accurate and reliable data to the users from time to time. By using the mobile application, the latest real-time information about the parking area can be obtained easily by the drivers. Therefore, this experiment has shown that objective 1 of this project has been achieved.

4.3 Experiments on the Evaluations of the Accuracy of Parking Vacancy Detection System

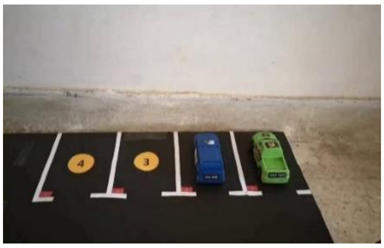
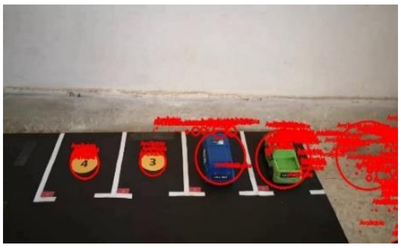
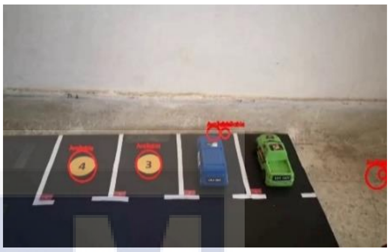
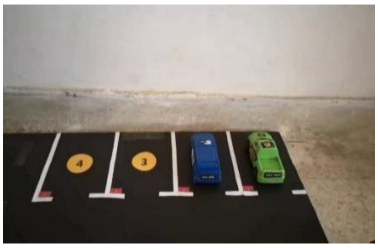
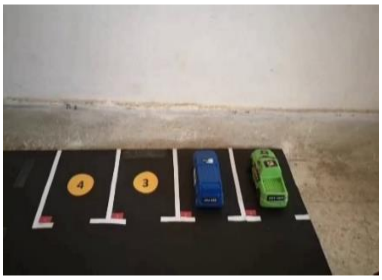
In this section, 3 experiments are conducted to evaluate the accuracy and reliability of the parking vacancy detection system. To detect the availability of car park, two computer vision algorithms, Haar Cascade Classifier and contour-detection method are used in this project. The accuracy of the contour parking vacancy detection system is evaluated by comparing the condition of the radius range of the detected area that are set in the python code. The results are shown in Table 4.2.

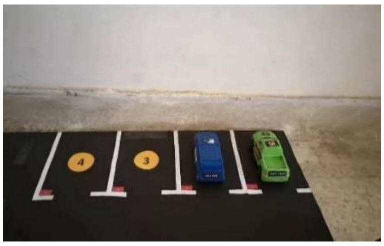
Next, the accuracies of the parking vacancy detection system using contour-detection algorithm and Haar Cascade Classifier are tested with three different types of parking, which are perpendicular parking, parallel parking and angle parking. The results are tabulated and recorded in Table 4.3, 4.4 and 4.5. Lastly, the accuracy of contour parking vacancy detection system and Haar Cascade parking vacancy detection system are evaluated under 4 different types of environment. The results are demonstrated in Table 4.6 and 4.7. The results are analyzed and discussed. The pros and cons of each computer vision algorithm are explained based on the experiments that have been done in this project.

4.3.1 Contour Parking Vacancy Detection with Different Settings of Radius Range

The experiment of the contour parking vacancy detection system with different settings of radius range is carried out in order to figure out the most accurate algorithm. The most ideal radius range obtained from this experiment is used to do further comparison in terms of the accuracy of parking vacancies detection with Haar Cascade Classifier. Six different settings of radius range are tested in this experiment to obtain the most ideal result. The six radius ranges are no radius range settings, 10 to 40 pixels radius range, 15 to 40 pixels radius range, 20 to 40 pixels radius range, 25 to 40 pixels radius range and 30 to 40 pixels radius range. To get a more reliable result, a reference condition is used to test with the six different radius range. All the results and the settings of radius range are recorded in Table 4.2 The further analyzation and interpretation of the results are done.

Table 4.2: The Experiment on the Contour Parking Vacancy Detection System with Different Settings of Radius Range

Settings of Radius Range (in pixels)	**Reference Condition**	**Results**
No Settings		
10-40		
15-40		
20-40		
25-40		

30-40		

From the results shown in Table 4.2, the efficiency of the contour parking vacancy detection system has improved when the radius range set in the system is getting smaller. When there are no settings to the radius range in the system, the result obtained is considered bad. From the result in case 1, more than 20 areas are detected by the system. This has proved that the detected areas with the same intensity or colors are bounded when the contour detection is used. In order to improve the contour parking vacancy detection system, the condition of radius range of detected area is added.

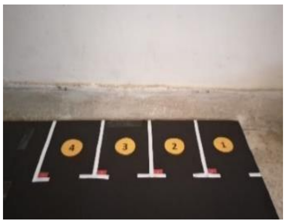
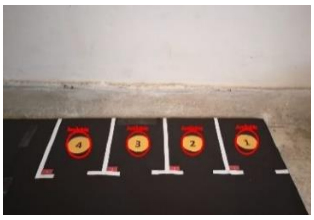
When the radius range is set between 10 and 40 pixels, the detected area has decreased to 6 areas only. This has proved that the condition of radius range of detected area is necessary to be included in this system. However, this result is not yet satisfied with the expected result. Therefore, the experiment is repeated by decreasing the radius range from 15-40 pixels to 30-40 pixels. When the radius is set in the range of 15-40 pixels, the detected areas is successfully decreased from 6 areas to 4 areas. Furthermore, the areas that are detected by the system are minimized to 3 areas when the radius ranges are set at 20 to 40 pixels and 25 to 40 pixels. When the radius range is reduced to 30 to 40 pixels, the result obtained is exactly same with the reference condition which only shows parking 3 and parking 4 are available to be parked. This has proved that the action of limiting the radius range of detected area works well in improving the accuracy of contour parking vacancy detection system.

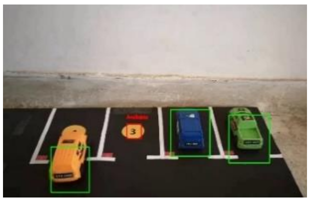
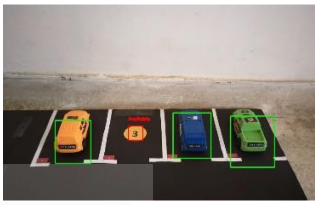
According to the results shown in Table 4.2, it has obviously demonstrated that the smaller the radius range that is set in the system, the more accurate the contour parking vacancy detection system in detecting the availability of car park. This experiment has shown that contour detection algorithm is a suitable method in identifying the availability of car park with the ideal radius range, 30-40 pixels. The program code with the condition of 30-40 pixels of radius range is used for the next section to compare with Haar Cascade Classifier.

4.3.2 Comparison of the Performance between Contour Parking Vacancy Detection System and Haar Cascade Parking Vacancy Detection System at Different Types of Parking

The evaluation on the performances of parking vacancy detection system is done using two methods: contour-detection method and Haar Cascade Classifier. Both methods used three types of parking which are perpendicular parking, angle parking and parallel parking. A camera is fixed on a tripod stand to make sure the distance and the height of the camera for both methods are at the same level. This is to obtain a more reliable result. These experiments are conducted to determine the most preferred method in detecting the availability of car parks. For each type of parking, five different cases are tested to evaluate the accuracy of the contour-detection method and Haar Cascade Classifier in detecting the status of car parks. This experiment is repeated for 20 times for each cases to get a more reliable result. The results are tabulated and recorded in Table 4.3, 4.4 and 4.5 together with its original image to have a better comparison. The findings are then analyzed and discussed.

Table 4.3: Parking Vacancy Detection System Using Contour-Detection Method and Haar Cascade Classifier at Perpendicular Parkings

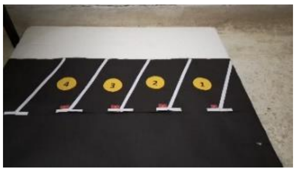
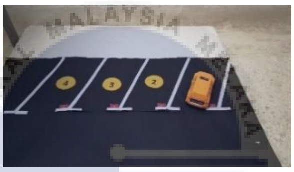
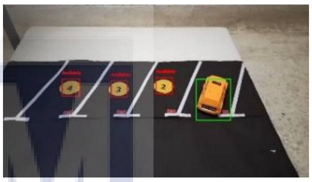
Case	Original Image	Results	
		Contour-Detection	**Haar Cascade**
1			
2			

3			
4			
5			

From the results shown in Table 4.3, the accuracy for both algorithms in detecting the availability of car parks is high. The results obtained for each case are exactly same with the original image. In this experiment, 3 toy cars with different colors which are yellow, blue and green are used. For case 1, when there are no cars parked at the parking lots, the status of the 4 car parks are displayed as available. When green and blue cars parked at the parking 1 and parking 2, the detection system showed only parking 3 and 4 are available. From the case 3 to 4, the results showed that the status of car park 4 is changed from available to not available once the yellow car has blocked the 30% of the yellow circle. When three cars parked at parking 1, 2 and 4 respectively, the result showed only car park 3 is available to be parked.

From the experiment, it has proved that the performances of contour parking vacancy detection system and Haar Cascade parking vacancy detection system will not be affected by the color of cars. In addition, the status of the car park will be changed from available to not available once the car blocked the 30% of the yellow circle in the middle of car park. Haar Cascade parking vacancy detection system is able to detect both cars and car parks whereas contour parking vacancy detection system can only detect the availability of parking. In summary, based on the results in Table 4.3, the accuracy of the contour parking detection system and Haar Cascade parking detection system in detecting the availability of car parks at perpendicular parking is 100% after repeating for 20 times. This is because both methods are able to provide accurate results for 5 different conditions. This experiment is repeated at the angle parking and parallel parking for obtaining more reliable evidences.

Table 4.4: Parking Vacancy Detection System Using Contour Detection Method and Haar Cascade Classifier at Angle Parkings

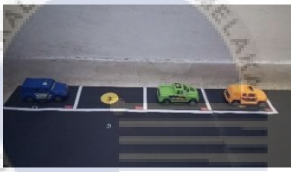
Case	Original Image	Results	
		Contour Detection	**Haar Cascade**
1			
2			
3			
4			
5			

The results shown in Table 4.4 have proved that the contour parking vacancy detection system and Haar Cascade parking vacancy detection system are able to identify the availability of the parking vacancies accurately whenever the conditions at the car park are changed. When there are no cars parked at the parking area, the status of each car park is labeled as available. In case 2, when the yellow car parked into the parking 1, the results for both detection systems showed parking 2, 3 and 4 area available only. After a few seconds, a blue car parked into parking 2, the results from both detection systems showed that the parking 2 is no longer be available. At this time, the results displayed only parking 3 and 4 are available to be parked. For case 4, the position of blue car is changed from parking 2 to parking 3. The contour parking vacancy detection system and Haar Cascade detection system detected the change, they showed that parking 2 is available but the parking 3 is occupied. For case 5, there are 3 cars parked at 3 parking lots: parking 1, 2 and 3 and only parking 4 is ready to be parked. The results shown by both detection system are similar with the original image.

From the observations, contour parking vacancy detection system and Haar Cascade parking vacancy detection system provided the same detection results for these 5 case. The accuracy for both detection systems to detect the availability of car parks at angle parking are 100% because both detection systems are able to give the accurate information even a sudden change is occurred. However, at case 4, the blue car is unable to be detected by Haar Cascade parking vacancy detection system. This is because the parking angle of the blue car in case 4 is different from other cases. Therefore, it has illustrated that if the parking style of the cars are different from the images that are used to train the Haar Cascade Classifier, it will not be detected by the Haar Cascade parking vacancy detection system. To improve the accuracy of car detection, the images of cars from different angles should be added in the positive folder during the Haar Cascade Classifier training.

Table 4.5: Parking Vacancy Detection System Using Contour Detection Method and Haar Cascade Classifier at Parallel Parkings

Case	**Original Image**	**Results**	
		Contour Detection	**Haar Cascade**
1			

2			
3			
4			
5			

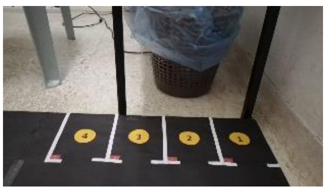
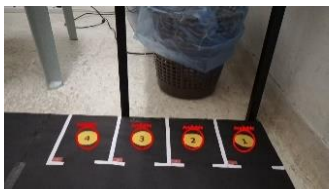
From the results tabulated in Table 4.5, it showed that both contour parking vacancy detection system and Haar Cascade parking vacancy detection system are able to provide the exact information of each cases. For the first case, the results obtained from both detection systems depicted all the 4 car parks are available because the yellow circles in the middle of car parks are not blocked. When the yellow car parked at parking 1, the status of the car park 1 is immediately changed to not available. For case 3 and 4, yellow car and green car parked at parking 1 and 2, the results showed that only parking 3 and 4 are available. From the observations in case 3 and 4, the accuracy of both detection systems in determining the availability of parking vacancy is not affected by the position of the cars as long as the cars have occupied at least 30% of yellow circle which placed at the middle of the car park. The performance of both detection systems at parallel parking is high and it has hit 100% of accuracy after repeating for 20 times.

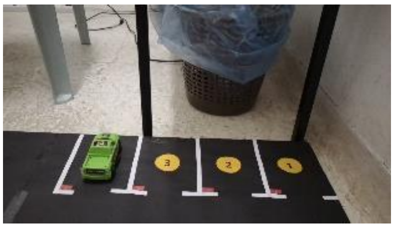
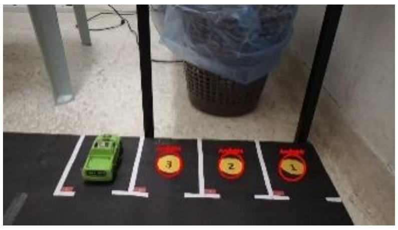
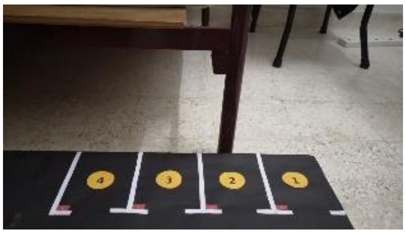
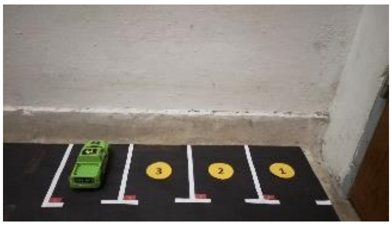
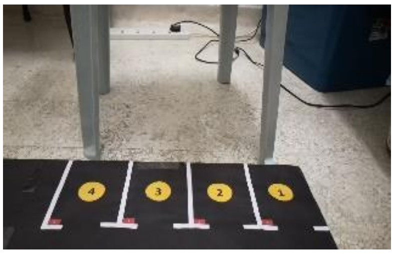
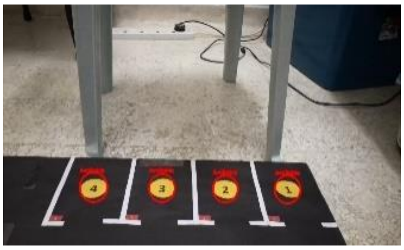
In conclusion, both contour parking vacancy detection system and Haar Cascade parking vacancy detection system performed well in detecting the status of the availability of car parks. This experiment is repeated at 3 different types of parking and each parking had 5 different cases. When the images of 15 different cases are processed with contour-detection method and Haar Cascade Classifier, the results obtained are exactly same with the original images. It concluded that the accuracy for both detection systems in detecting the status of the availability of the car parks is 100%. This proved that the performance of both detection systems is not affected by colors of cars, position of the cars and types of parking. In this project, the detection of cars are not necessary to be included. Hence, the flaws of Haar Cascade parking vacancy detection system in detecting the presence of cars can be neglected. An experiment for comparing the performance between both detection systems at different environments is carried out in subchapter 4.3.3 to get more reliable results.

4.3.3 Comparison of the Performance between Contour Parking Vacancy Detection System and Haar Cascade Parking Vacancy Detection System at Different Environments

Since the performance of both parking vacancy detection system is the same when evaluated at three different types of parking, an experiment is conducted at different environments. An environment is chosen as reference environment for Haar Cascade Classifier training. This experiment is carried out at 4 different environments and for each environment, 2 different cases are tested. A camera is fixed on a tripod stand to make sure the distance and the height of the camera for different environments are at the same level. This is to obtain a more reliable result. The results are tabulated and recorded in Table 4.6 and 4.7 together with its original image to have a better comparison. The findings are then analyzed and discussed.

Table 4.6: Results of the Contour Parking Vacancy Detection System at Different Environment

Environment	Original Image	Result
Reference		

		
1		
		
2		
		
3		

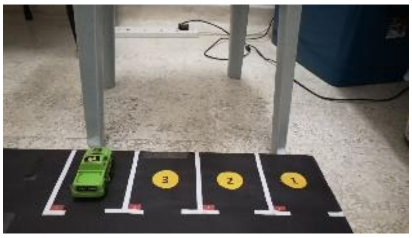
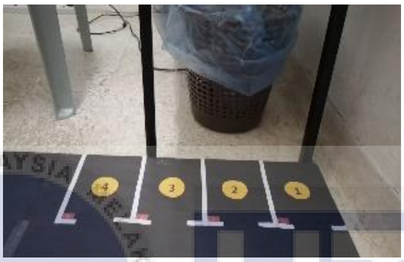
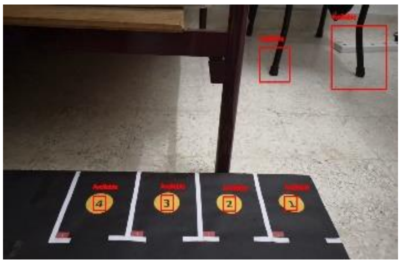
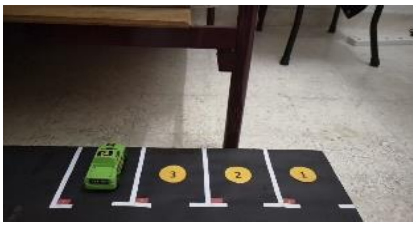
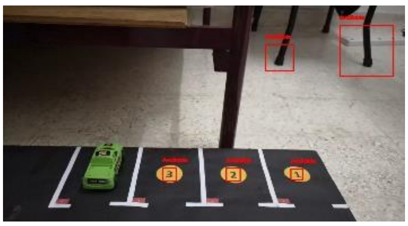
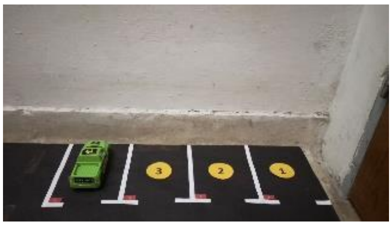
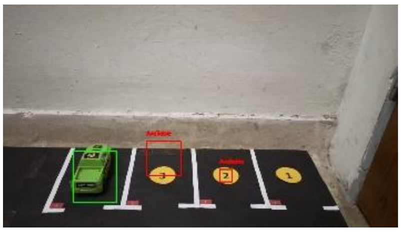
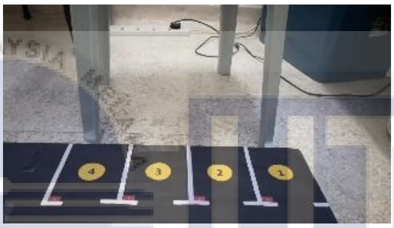
		

Table 4.7: Results of the Haar Cascade Parking Vacancy Detection System at Different Environment

Environment	Original Image	Result
Reference		
		
1		
		

2		
		
3		
		

From the observations in Table 4.7, the results obtained from the contour parking vacancy detection system showed that parking 1, 2, 3 and 4 are available when there are no cars parked at the car parks. However, when the green car parked at the parking 4, the results showed only parking 1, 2 and 3 are available. Based on the results shown in Table 4.7, the accuracy of contour parking vacancy detection system is 100% because 8 out of 8 cases have the exactly same information with the original images. This is because it performed well in detecting the availability of car parks at the 4 different environments.

In contrast, the performance of Haar Cascade parking vacancy detection system at different environments is unsatisfied. When the experiment is done at reference environment, the results obtained are the same with the original images. However, when the environment condition is changed, few detection errors (other than car parks) are occurred. This caused the users receiving inaccurate information. In the second environment, legs of chairs are detected as available car parks by the Haar Cascade parking vacancy detection system. For the third environment, the status of parking 1 showed not available even though there is no car parked at the parking. For the forth environment, there is an error detected by Haar Cascade parking vacancy detection system. This is because the detected error is not included into the negative folder of the trained Haar Cascade Classifier. Hence, the accuracy of Haar Cascade parking vacancy detection system in detecting the availability of car parks at different environments is 25% because only 2 out of 8 cases are detected correctly without any errors.

In summary, the accuracy of contour parking vacancy detection system is higher than the Haar Cascade parking vacancy detection system when the experiment is conducted at different environments. It showed that Haar Cascade algorithm is a precise method for a specific condition. However, the performance of the Haar Cascade parking vacancy detection system is affected by the condition of surroundings such as chair, plug, window and etc. Table 4.8 shows the pros and cons of contour-detection method and Haar Cascade Classifier.

Table 4.8 : The Pros and Cons of Contour-detection method and Haar Cascade Classifier

	Contour-detection Method	Haar Cascade Classifier
Pros	▪ Suitable to be implemented for most of the conditions such as different types of parking and different surrounding environments. ▪ Simple computer vision algorithm.	▪ High performaces for specific condition. ▪ Can train more that one object detection algorithms such as cars and car parks
Cons	▪ Many small areas are detected by contour-detection method if the condition of radius range is not added in the program code.	▪ Complex computer vision algorithm. ▪ Consuming time on training of Haar Cascade Classifier ▪ Accuracy is dropped when it is applied at different surrounding environment.

4.4 Summary

In sum, the experiment done in section 4.2 has achieved objective 1 whereas the experiments done in section 4.3 has evaluated the performance of the parking vacancy detection system in terms of accuracy. Table 4.9 depicts the summary of the experiments. The experiment done in section 4.2 proved that the information at the parking area has been successfully transferred to Firebase and mobile application using IoT ecosystem. The mobile app's users are able to know the latest real-time parking vacancy status at anytime and anywhere. For experiments done in section 4.3, two computer vision algorithms, contour-detection method and Haar Cascade are used. After conducting an experiment with six different radius ranges, the most optimized radius range for contour-detection method is 30-40 pixels. Next, the performance of both contour parking vacancy detection system and Haar Cascade parking vacancy detection system are high when tested at different types of parking. However, the accuracy of Haar Cascade parking vacancy detection system is dropped to 25% when it is used to detect the availability of car parks at different environments. The results obtained showed that contour parking vacancy detection system has a consistent performance whenever at different types parking or at different environments. The IP camera used in this experiment is able to detect 4 parking lots per detection due to the distance between camera and parking and height of tripod stand that are set.

Table 4.9 : Summary of Experiments

Experiments	Objective 1	Objective 2
Real-time Data Synchronization between OpenCV, Firebase and Mobile Application	✓	
Experiment on the Contour Parking Vacancy Detection System with Different Settings of Radius Range		✓
Comparison of the Performance between Contour Parking Vacancy Detection System and Haar Cascade Parking Vacancy Detection System at Different Types of Parking		✓
Comparison of the Performance between Contour Parking Vacancy Detection System and Haar Cascade Parking Vacancy Detection System at Different Environments		✓

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Introduction

This chapter discusses the overall conclusions about Final Year Project (FYP). This chapter consists of conclusion of current progress of this project and recommendations for the further works.

5.2 Conclusion

This thesis consists of 5 chapters which are introduction, literature review, methodology, the preliminary results and the conclusion. In Chapter 1, the background study, motivations, problem statements of the current parking system, objectives of the project and the scope of the project are discussed. The development of advanced technologies and techniques, the consuming of time and cost in searching a parking lot, the negative impacts that are caused by the burning of additional fuel are the motivations to developed an effective parking system. The objectives of the project is determined based on the limitations of the current on-street parking system in Malaysia. The objectives of this project are to design and develop an on street parking system with the integration of mobile application for providing real-time information to the users and to evaluate the performance of system in terms of accuracy in detecting vacancy for car parks. The scope of the project is also set to the components selected, the conditions of lighting, the prototype and software used.

In Chapter 2, the reviews on the existing on-street parking system are done by referring the trusted resources such as articles and journals that related to this project. The performance of the existing on-street parking system is evaluated and compared based on the sensing method, the accuracy of detection of parking availability, the number of sensor per lot and the limitations of the system. The sensing method that are used in the existing on-street parking system are magnetic sensor, photoelectric sensor, ultrasonic sensor, camera, mobile sensing unit and accelerometer. After that, camera is chosen as the sensing method to detect the parking availability. Camera as sensor can provide the real-time information and exact position of parking lots to the drivers. In addition, the camera can improve the security of the on-street parking area. The comparison and justification for the computer vision algorithms are done to choose the two most preferred method in order to apply in this project. A mobile application with the function of checking the parking availability is proposed to improve the flaws of the on-street parking system.

The methodology of this project is discussed in Chapter 3. A prototype is built with tripod stand, an IP camera, toy cars, 4 perpendicular parking lots, 4 angle parking lots and 4 parallel parking lots. This proposed system consists of three major components which is sensing part (camera), the server (Intel UP Squared board) and the display platform (mobile application and cloud). The images that are captured by the IP camera is sent to Intel UP Squared board to undergo image processing step to identify the status of parking availability for each parking lot. All the inputs and outputs of the system are stated and explained clearly in the section 2.1. The design flowchart of the mobile application is also explained. The mobile application is developed using Android Studio software. The pop-out notification function is provided to remind the users about the updated information. The steps for conducting the experiments on the evaluation of the on-street parking system for the performances of IoT ecosystem and parking vacancy detection system are explained in details. The observations and results are discussed in the Chapter 4.

In Chapter 4, the observations and results from the experiments done are shown in table. In section 4.2, the results obtained prove that the real-time data synchronization between OpenCV, Firebase and mobile application is done. The experiment is repeated for 10 times to get a reliable result. The experiment has achieved objective 1 of this project. Next, in section 4.3, three different experiments are carried out to evaluate the performance of both contour parking vacancy detection system and Haar Cascade parking vacancy detection system in terms of accuracy in detecting the status of availability of car parks. The contour-detection method with six different settings of radius range are done to determine the most ideal radius range. After conducting the experiment, the most ideal radius range is 30-40 pixels. The experiments on the comparison between contour parking vacancy detection system and Haar Cascade parking vacancy detection system are conducted at three different types of parking and also at 4 different environments. The results obtained shows that both detection systems achieves 100% of accuracy in detecting the parking vacancy at different types of parking after repeating each case for 20 times. However, when both detection systems are implemented at different environments, contour parking vacancy detection system has a higher accuracy than the Haar Cascade parking vacancy detection system. These experiment has achieved objective 2 of this project. Lastly, the summary of the current work is discussed in Chapter 5. A brief overview for each chapter is done and the recommendations for improving this project are also stated in this chapter

5.3 Recommendations

The improvement on the Haar Cascade parking vacancy detection system should be done in the future. This is because a high accurate algorithm is necessary to make sure that the users always get the correct information. The recognization of car plate can also be done in the future after improving the Haar Cascade parking vacancy detection system. The functions of reserving parkings and online payment can be added to this system. This will make this microprocessor-based on-street parking notification system more user-friendly.

REFERENCES

[1] Faheem, S. A. Mahmud, G. M. Khan, M. Rahman, and H. Zafar, "A survey of intelligent car parking system," *Journal of Applied Research and Technology*, vol. 11, no. 5, pp. 714–726, 2013.

[2] M. Aazam, I. Khan, A. A. Alsaffar, and E. N. Huh, "Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved," *Proceedings of 2014 11th International Bhurban Conference on Applied Sciences and Technology, IBCAST 2014*, pp. 414–419, 2014.

[3] L. Kwan, "Double-Parked Car Gets Rammed by Angry M'sian Driver After He Honked for 30 Mins - WORLD OF BUZZ," *World of Buzz*, 2018. [Online]. Available: https://www.worldofbuzz.com/double-parked-car-gets-rammed-by-angry-msian-driver-after-he-honked-for-30-mins/.

[4] Kathleen Michael, "Chaotic situation at Subang Jaya business hub - Metro News | The Star Online," *Star Online*, 2019. [Online]. Available: https://www.thestar.com.my/metro/metro-news/2019/05/13/chaoticsituation-atsubang-jayabusiness-hub/.

[5] "Which of these parking peeves bug you the most?" *The Star Online*, 2017. [Online]. Available: https://www.thestar.com.my/news/nation/which-of-these-parking-peeves-bug-you-the-most/.

[6] F. Perera, "Pollution from fossil-fuel combustion is the leading environmental threat to global pediatric health and equity: Solutions exist," *International Journal of Environmental Research and Public Health*, vol. 15, no. 1, 2018.

[7] "Malaysia on track to realise smart cities vision," *Bernama*, 2018. [Online]. Available: https://www.edgeprop.my/content/1419207/malaysia-track-realise-smart-cities-vision-says-ict-leader.

[8] "Mobile Operating System Market Malaysia," *Stat Counter*, 2018. [Online]. Available: http://gs.statcounter.com/os-market-share/mobile/malaysia.

[9] A. Khanna and R. Anand, "IoT based smart parking system," *2016 2016 International Conference on Internet of Things and Applications, IOTA 2016*, no. November, pp. 266–270, 2016.

[10] A. R. Y. Aryzky, "Intelligent parking system," no. May, pp. 160–167, 2010.

[11] C. Roman, R. Liao, P. Ball, S. Ou, and M. De Heaver, "Detecting On-Street Parking Spaces in Smart Cities: Performance Evaluation of Fixed and Mobile Sensing Systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2234–2245, 2018.

[12] V. Coric and M. Gruteser, "Crowdsensing maps of on-street parking spaces," *Proceedings - IEEE International Conference on Distributed Computing in Sensor Systems, DCoSS 2013*, pp. 115–122, 2013.

[13] D. Acharya, W. Yan, and K. Khoshelham, "Real-time image-based parking occupancy detection using deep learning," *CEUR Workshop Proceedings*, vol. 2087, pp. 33–40, 2018.

[14] F. Html and Z. Zhang, "Cópia - A Street Parking System Using Wireless Sensor Networks," vol. 2013, no. 2013, pp. 5–11, 2014.

[15] H. Zhu and F. Yu, "A Vehicle Parking Detection Method Based on Correlation of Magnetic Signals," *International Journal of Distributed Sensor Networks*, vol. 2015, 2015.

[16] R. Grodi, D. B. Rawat, and F. Rios-Gutierrez, "Smart parking: Parking occupancy monitoring and visualization system for smart cities," *Conference Proceedings - IEEE SOUTHEASTCON*, vol. 2016-July, pp. 1–5, 2016.

[17] B. Xu, O. Wolfson, J. Yang, L. Stenneth, P. S. Yu, and P. C. Nelson, "Real-time street parking availability estimation," *Proceedings - IEEE International Conference on Mobile Data Management*, vol. 1, pp. 16–25, 2013.

[18] C.-H. L. C.-H. Lee, M.-G. W. M.-G. Wen, C.-C. H. C.-C. Han, and D.-C. K. D.-C. Kou, "An automatic monitoring approach for unsupervised parking lots in outdoors," *Proceedings 39th Annual 2005 International Carnahan Conference on Security Technology*, 2005.

[19] V. Kepuska and H. Alshamsi, "Smart Car Parking System," no. August 2016, 2017.

[20] J. A. Vera-Gómez, A. Quesada-Arencibia, C. R. García, R. S. Moreno, and F. G. Hernández, "An intelligent parking management system for Urban areas," *Sensors (Switzerland)*, vol. 16, no. 6, pp. 1–16, 2016.

[21] S. Banerjee, P. Choudekar, and M. K. Muju, "Real time car parking system using image processing," *ICECT 2011 - 2011 3rd International Conference on Electronics Computer Technology*, vol. 2, pp. 99–103, 2011.

[22] H. R. H. Al-Absi, J. D. D. Devaraj, P. Sebastian, and Y. V. Voon, "Vision-based automated parking system," *10th International Conference on Information Sciences, Signal Processing and their Applications, ISSPA 2010,* no. June, pp. 757–760, 2010.

[23] Z. Zhang, X. Li, H. Yuan, and F. Yu, "A Street Parking System UsingWireless Sensor Networks Zusheng," *2008 International Symposium on Collaborative Technologies and Systems*, vol. 5, no. 12, pp. 48–57, 2008.

[24] H. R. H. Al-absi, J. Dinesh, and D. Devaraj, "Vision-Based Automated Parking System," no. June, 2010.

[25] P. Meduri and E. Telles, "A Haar-Cascade classifier based Smart Parking System," pp. 66–70.

[26] Aparna Vashishtha and Himanshu Sharm, "Artificial Intelligence API for Car Parking Management System," *electronicsforu.com*, 2018. [Online]. Available: https://electronicsforu.com/electronics-projects/prototypes/artificial-intelligence-api-car-parking-management-system.

[27] C. S. Tutika, C. Vallapaneni, and B. Karthikeyan, "Image Handling and Processing for Efficient Parking Space Detection and Navigation Aid School of Electronics Engineering."

[28] S. Ma, H. Jiang, M. Han, J. Xie, and C. Li, "Research on automatic parking systems based on parking scene recognition," *IEEE Access*, vol. 5, pp. 21901–21917, 2017.

[29] S. F. Lin, Y. Y. Chen, and S. C. Liu, "A vision-based parking lot management system," *Conf. Proc. - Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 2897–2902, 2007.

[30] X. Y. Gong, H. Su, D. Xu, Z. T. Zhang, F. Shen, and H. Bin Yang, "An Overview of Contour Detection Approaches," *International Journal of Automation and Computing*, vol. 15, no. 6, pp. 656–672, 2018.

[31] S. Guennouni, A. Ahaitouf, and A. Mansouri, "A Comparative Study of Multiple Object Detection Using Haar-Like Feature Selection and Local Binary Patterns in Several Platforms," *International Journal of Automation and Computing*, vol. 2015, pp. 1–8, 2015.

APPENDICES

APPENDIX A GANTT CHART FOR FYP 1

Project Activity	Sept'18				Oct'18				Nov'18				Dec'18	
FYP1	Weeks													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Title Discussion and Selection	■	■												
Research and Journal Findings	■	■	■	■	■	■	■	■	■	■	■			
Motivation & Problem Statement		■	■											
Objectives & Scope			■	■										
Literature Review			■	■	■	■	■	■						
Components Selection						■	■	■						
Preliminary Hardware Development								■	■	■				
Methodology -Experiment Designed							■	■	■	■				
Preliminary Results										■	■			
Submission for Draft of Progress Report											■			
FYP 1 Seminar												■		
Finalizing the Progress Report													■	■
Submission for Final Report														■

APPENDIX B GANTT CHART FOR FYP 2

Project Activity	Feb-19		March-19				April-19				May-19			
FYP 2	Weeks													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Project Development	■	■	■	■	■	■								
Android App Development	■	■	■	■										
Integration between hardware and Intel UP Squared board				■	■									
Integration between Android Application and Firebase						■	■							
Integration between OPENCV and Firebase						■	■	■						
Conduct Experiments								■	■	■	■			
Presentment of Results											■	■		
Analysis of Results											■	■		
Report Compiling and Draft Submission											■	■	■	
FYP2 Presentation														■
Submission of Final Report														■

APPENDIX C COUNTOUR-DETECTION METHOD's CODE

```
#import the necessary packages
import cv2
import numpy as np
from scipy import misc
from PIL import Image
import time
from imutils import contours
from skimage import measure
import imutils
from firebase import firebaseprint(cv2.__version__)


firebase = firebase.FirebaseApplication('https://parkingsystem-
innovate123.firebaseio.com/')
video_src='car detection/1.jpg'
cap=cv2.VideoCapture(video_src)

while True:
  a=0
  parking1 = 0
  parking2 = 0
  parking3 = 0
  parking4 = 0
  ret,img=cap.read()
  if (type(img) == type(None)):
    break

    hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
    lower_yellow=np.array([10,100,100])
    upper_yellow=np.array([150,255,255])
    mask=cv2.inRange(hsv,lower_yellow,upper_yellow)
    cnts = cv2.findContours(mask.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    cnts = imutils.grab_contours(cnts)
    cnts = contours.sort_contours(cnts)[0]

# loop over the contours
    for (i, c) in enumerate(cnts):
# draw the bright spot on the image
          (x, y, w, h) = cv2.boundingRect(c)
          ((cX, cY), radius) = cv2.minEnclosingCircle(c)
```

```
if (radius>30 and radius<40):
   cv2.circle(img, (int(cX), int(cY)), int(radius),(0, 0, 255), 3)
   print ((cX,cY))
  cv2.putText(img, "Available", (x, y - 15))
  cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0, 0, 255), 2)
  if( cX >= 660 and cX <= 760 and cY >= 350 and cY <= 500):
     parking1 = 1
     print("1: available")
  elif( cX >= 350 and cX <= 450 and cY >= 350 and cY <= 500):
     parking3 = 1
     print("3: available")
  elif( cX >= 500 and cX <= 600 and cY >= 350 and cY <= 500):
     parking2 = 1
     print("2: available")
  elif( cX >= 180 and cX <= 280 and cY >= 350 and cY <= 500):
     parking4 = 1
     print("4: available")
  else:
     print(" no parking found. ")

  if( parking1 == 1 ):
     firebase.put('Parking','1','parking 1: available')
  else:
     firebase.put('Parking','1','parking 1: not available')
  if( parking2 == 1 ):
     firebase.put('Parking','2','parking 2: available')
  else:
     firebase.put('Parking','2','parking 2: not available')
 if( parking3 == 1 ):
     firebase.put('Parking','3','parking 3: available')
 else:
     firebase.put('Parking','3','parking 3: not available')
if( parking4 == 1 ):
     firebase.put('Parking','4','parking 4: available')
else:
     firebase.put('Parking','4','parking 4: not available')

# show the output image
   cv2.imshow("Image", img)
   cv2.imwrite('result.jpg',img)
   cv2.waitKey(0)
```

APPENDIX D HAAR CASCADE CLASSIFIER's CODE

```
#import the necessary packages
import cv2
import time
from firebase import firebaseprint(cv2.__version__)

firebase = firebase.FirebaseApplication('https://parkingsystem-
innovate123.firebaseio.com/')
firebase = firebase.FirebaseApplication('https://parkingsystem-
innovate123.firebaseio.com/')

cascade_src = 'car_cascade (19).xml'
cascade_src1 = 'cascadeparking10.xml'
video_src = 'car detection/5.jpg'
#video_src = 'dataset/video1.avi'

cap = cv2.VideoCapture(video_src)
car_cascade = cv2.CascadeClassifier(cascade_src)
parking_cascade = cv2.CascadeClassifier(cascade_src1)
#img1 = cv2.VideoCapture('car detection/22.jpg')

while True:
  a=0
  parking1 = 0
  parking2 = 0
  parking3 = 0
  parking4 = 0
  ret,img=cap.read()
  if (type(img) == type(None)):
    break

    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    parking = parking_cascade.detectMultiScale(gray, 1.1, 1)
    for (x,y,w,h) in parking:
       cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,255),2)
      cv2.putText(img, "Available", (x, y - 15),
      cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0, 0, 255), 2)
      #print(str(x)+","+str(y))
      if( x >= 630 and x <= 700 and y >= 300 and y <= 400):
          parking1 = 1
          print("1: available")
```

```
elif( x >= 300 and x <= 400 and y >= 300 and y <= 400):
        parking3 = 1
        print("3: available")
elif( x >= 450 and x <= 550 and y >= 300 and y <= 400):
        parking2 = 1
        print("2: available")
elif( x >= 150 and x <= 250 and y >= 300 and y <= 400):
        parking4 = 1
        print("4: available")
else:
        print(" no parking found. ")
#time.sleep(3)
 cars = car_cascade.detectMultiScale(gray, 1.1, 1)
 for (ex,ey,ew,eh) in cars:
      cv2.rectangle(img,(ex,ey),(ex+ew,ey+eh+20),(0,255,0),2)

      if( parking1 == 1 ):
          firebase.put('Parking','1','parking 1: available')
      else:
          firebase.put('Parking','1','parking 1: not available')
      if( parking2 == 1 ):
          firebase.put('Parking','2','parking 2: available')
      else:
          firebase.put('Parking','2','parking 2: not available')
      if( parking3 == 1 ):
          firebase.put('Parking','3','parking 3: available')
       else:
          firebase.put('Parking','3','parking 3: not available')
        if( parking4 == 1 ):
          firebase.put('Parking','4','parking 4: available')
        else:
          firebase.put('Parking','4','parking 4: not available')

        cv2.imshow('img', img)
        cv2.imwrite('result.jpg',img)

         cv2.waitKey(0)
        if cv2.waitKey(33) == 27:
            cv2.destroyAllWindows()
            break
```