REAL TIME GESTURE RECOGNITION SYSTEM FOR ADAS

BY

CHEE YING XUAN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfilment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONS)

INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology

(Perak Campus)

MAY 2018

DECLARATION OF ORIGINALITY

I declare that this report entitled "**REAL-TIME GESTURE RECOGNITION SYSTEM FOR ADAS**" is my own work except as cited in the references. The report not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ___________________________

Name : ___________________________

Date : ___________________________

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deep gratitude to my project supervisor Dr Lau Phooi Yee who provides me this interesting, intuitive and challenging topic for my final year project and guided me throughout the whole project period. Whenever I am encountering difficulties in the project development, she is always there for providing encouragement, motivation, useful idea, advice and feedback as well with great patience. I could not asked for a better supervisor in my university life.

Besides that, I would like to present very special thanks to my project moderator Ms Lai Siew Cheng who give me an opportunity to express and present my final year project idea. I am so much appreciate her effort in evaluating the quality of my final year project and providing valuable feedback for me to achieve a better improvement.

In addition, I would like to thank my academic advisor Mr Amir Amin who willing to scarify his precious time to advise me every time when I am facing difficulties no matter on academic, extracurricular and personal issues as well. He is the one who always tracking on my academic performance and provides useful information on how to study well in order to achieve better result.

Last but not least, I wish to thank my parent who raise me and guide me to become a better person. I am so much appreciate them for providing me an opportunities to pursue a higher education in UTAR and provides me encouragement whenever I am about to give up. I may not be able to finish my final year project without their support.

ABSTRACTS

The real-time gesture recognition system is developed accordance with the objective of ADAS which is to make cars safer to drive and assist driver in the driving process. The system is aimed to simplify and enhance the interaction between human and computer by implementing the vision-based technique that didn't requires any complex sensor device in collecting user hand gesture as an input for gesture recognition. It allow people convey their actions or intentions by using natural mid-air hand gesture to interact with the infotainment system function. Eventually, the system will be developed to track and recognize several human static hand gestures by implementing sets of image processing techniques and algorithms that have been developed throughout the system development process. The system process is separated into five stages which include image acquisition, background subtraction, hand segmentation, features extraction and gesture recognition. Various diagrams are used to describe to overall system design which include block diagram, use-cases diagram and activity diagrams. The Evolutionary Prototyping methodology is also used to speed up the system development process and improve the quality of the final system through several refinement on the prototype. High-level Python programming language is used to develop the system as it provides easy syntax that allow quick coding and provides various standard libraries which enable the execution of complex functionalities easily. OpenCV open source library also being used for its various functions that related to object tracking and image processing. Eventually, the system functionality, average recognition rate, accuracy and misclassification rate of the system is being evaluated in the system testing through functional testing and non-functional testing which include black-box testing, system performance testing and classification performance testing. The weaknesses of the system will be recorded as part of the future work of the project in order to achieve better improvement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADAS	*Advanced Driver Assistance System*
HCI	*Human-Computer Interaction*
CV	*Computer Vision*
RGB	*Red Green Blue*
RGB-D	*Red Green Blue-Depth*
HSV	*Hue Saturated Value*
GPS	*Global Positioning System*
ToF Camera	*Time-of-Flight Camera*
DTW	*Dynamic Time Wrapping*
HMM	*Hidden Markov Model*
2D	*Two dimension*
3D	*Three dimension*
UI design	*User Interface design*
UML Diagrams	*Unified Modelling Language Diagrams*
IDE	*Integrated Development Environment*
OpenCV	*Open Source Computer Vision*
RAM	*Random-Accessed Memory*
ARR	*Average Recognition Rate*
TP	*True Positive*
FP	*False Positive*
CNN	*Convolutional neural network*
SVM	*Support Vector Machine*

CHAPTER 1: INTRODUCTION

1.1Problem Statement and Motivation

In the past, people used to control of their vehicle infotainment system manually by all their hand on the physical controllers, buttons or even on a touch screen interface such as answer or reject incoming phone call, volume control of the audio system, temperature control and so forth. When a person perform these actions, it can take their attention away from driving in operate the system and lead to the happen of car accident. According to the report from (Waterdown Collision, 2017), one of the leading causes of car accident is distracted drivers and somewhere between 25-50 percent of all motor vehicle crashes in the U.S. are directly related to driver distraction as the root cause of automobile accidents. A second of distraction in driving may lead to serious impact and pose a threat to driver himself and other road user. ‘Based on recent studies, anything that takes your attention away, any glance away from the road for two seconds or longer can increase the risk of an accident from four to 24 times’, said Dr David Hurwitz from the Oregon State University (Oregonstate.edu, 2017).

In order to reduce the rate of road accidents that cause by driver distraction, the automotive industry looking forward for a solution to simplify and enhance the interaction between human and computer (HCI) which allow the driver to interact with the vehicle infotainment system using natural mid-air hand gesture while remaining fully focus on the road. With the fast growing development in ADAS technologies, HCI has become the most important application fields of computer vision with the main goal of using an intuitive and effortless ways to interact with the computer without physically interact with the touch screen interface, buttons and controller. In this case, hand gesture interaction may be the great alternative for the driver interacting with the vehicle infotainment system with natural hand gestures. By using specific image processing algorithms and techniques, the detected hand gesture is being recognized and generate corresponding instruction as functional input to control the in-vehicle system function. In this way driver no need to take their attention away from driving while operating the infotainment system.

1.2 Background Information

This section will mainly focus on the background information that related to gesture recognition and image processing techniques and algorithms that will be involved in this project.

Advance driver assistance system (ADAS)

Advance driver assistance system (ADAS) are the systems that developed with the objective of making cars safer to drive and assist driver in the driving process by enhance and automate the vehicle system in order to provide a safety and comfortable driving experience thus reduce the rate of accident that cause by driver negligent (Partners, 2016). Technavio's latest shows that gesture-based interfaces is one of the top three emerging trends that driving the global ADAS market (Business Wire, 2016). The gesture recognition systems have enter the automotive industry with the aim of provide a safety, comfort and convenience driving experience to drivers in controlling the vehicle infotainment system functions with only nature hand gestures and without losing control on the steering wheels. These functions include answer or reject incoming phone call, audio system control, temperature control as well as the GPS navigation control.

Human-Computer Interaction

Human-Computer interaction (HCI) refers to the interaction between a human and computer using some communicating median such as mouse, keyboard, joystick, button, touch screen interface and so forth. It aimed to simplify and enhance the interaction between human and computer. With the growing of HCI technology in recent years, large number of researchers began to create intuitive and convenient Computer Vision based HCI systems in several application area and it has generated more and more attention. This type of HCI system has conceals a great potential for the board application in gesture recognition, object tracking, visual virtual control and so forth.

Computer Vision

Computer Vision (CV) is the high level process that involved how computer automatically extract, analysis and understand the useful information from an image or a video sequence. The automatic visual understanding can be achieved by the development of theoretical and algorithmic basis and it seeks to automate the tasks that can be accomplish by the human visual system.

Gesture

Gesture is a form of non-verbal communication which involve in human everyday interaction and it assist people in conveying the meaning of an action or word especially for people with hearing and speech impairment problem.

Static and Dynamic Hand Gesture

Generally, there are two concept of hand gesture need to be differentiated in the hand gesture analysis which is the static hand gesture and the dynamic hand gesture (Marilly, et al., 2013). Static hand gesture also known as hand posture doesn’t change in the period of recognition process and another concept refers to the dynamic hand movement which the temporal trajectory of some estimated parameter over time (Jolliffe, 2002) or as sequences of hand posture (Chen, 2008).

Gesture Based Interaction

Gesture based interaction in HCI allow people convey their actions or intentions by using natural mid-air hand gesture to interact with the system function. Although gestures can easily detect and recognize by human, but it is challenging when it comes to implement an automatic approach due to a number of constraints in capturing gesture and a wide semantic gaps that refer to the inconsistent between the user interpretation and the extracted information from visual data.

Hand Gesture Recognition

Gesture recognition refers to the tracking and recognition of human movements that occur in different parts of body such as head, body, arm as well as facial expression (Hofmann, et al., 1998). It can be done in either two dimensional or three dimensional. There is basically two approaches of hand gesture recognition which is device-based approaches and vision-based gesture approaches.

Device-Based Approaches

Device-based techniques also known as marker-based techniques use a sensor device such as data glove to collect the motion of palm and fingers as the system input. But this approaches are usually more costly and require consume more resources on additional setup or calibration steps.

Figure 1.2-F1 Leap Motion (Akhtar, 2015)

Vision-Based Approaches

Vision-based techniques (Wachs, et al., 2011) is the markerless approaches which involve only camera or sensor as the system input such as Kinect and Time of Flight (ToF). It can be split into two categories which include 3D model-based approaches and appearance based approaches.

Figure 1.2-F2-Kinect Sensor (Amos, 2011)

3D Model-Based Approaches

The 3D model-based approaches are the high level approaches that describe hand postures and movement by tracking and modelling the entire articulated hand model in 3D and incorporate hand shape constraints easily. However, this approaches need to concern with the high accuracy post estimation algorithms (yao & Fu, 2014). It is too computational expensive to be run in real-time in term of processing time and model initialization sensitivity (Marilly, et al., 2013), (Rossol, et al., 2016), (Euda, et al., 2003).

Appearance Based Approaches

The appearance-based approaches also known as view-based approaches model the appearance of hand by extracting hand features from the captured image such as colour, area, silhouette, contour, pixel flow and so forth. This approach often sensitive to background clutter, lightning conditions, skin colour and movement speed. Consider on various elements to enable real-time interaction such as processing time, computational cost, algorithm complexity and resource limitation, appearance-based gesture recognition is more preferable for this project.

Image Processing

In order to perform gesture recognition, image processing is the crucial step in processing the raw video sequence before the image information being analyse and extract the required information for further processing. It comprise of several processes which include background subtraction, image segmentation and features extraction.

Background Subtraction

Background subtraction is a common image processing technique for removing the unnecessary background, noise that associated with the image and generate a foreground mask which is the region of interest that require to be extracted from the image. For this project, the BackgroundSubtractorMOG2 from OpenCV library (OpenCV dev team, 2014) will be implemented for the background subtraction process in extracting the foreground mask which is the user’s hand from the background model and used for the next step.

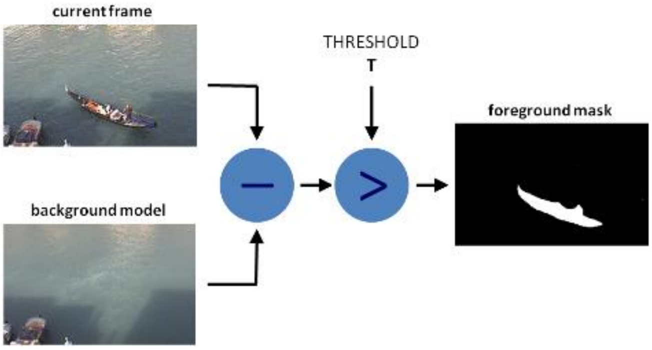


Figure 1.2-F3 Background Subtraction (Doxygen, 2018)

Image Segmentation

Image segmentation is the process of separation between the region of interest and unwanted segments in pixel form in order to make the image meaningful and easier to be analysed. For this project, the region of interest that need to be extracted from the image is the hand region which is the segmentation between skin and non-skin pixels. To achieve the expected result, colour space conversion has to be performed for better representation of colour in extracting hand region.

Figure 1.2-F4 Hand Segmentation (Blundell, 2011)

RGB Colour Space

RGB which representing red, green and blue is one of the common colour space that used by most image capturing devices for storing and processing digital image data. It involves mixing of Chrominance and Luminance information (Shaik, et al., 2015) which is not preferable for colour based detection and analysis. Therefore, it has to be transformed into another preferable colour space which is HSV colour space for this case.

HSV Colour Space

HSV colour space defines colour portion (Hue) in terms of degree of grey in colour (Saturation) and its brightness value (Value) which is more close to human perception colour (Bear, 2018). It is a simpler colour space that involves lesser colour pixel which is more suitable for representing the image in detecting skin pixel within the predefined colour range.

Image Thresholding

Threshold refers to assignation of pixel value to either black or white based on the threshold value. OpenCV library has provides the threshold function for extracting the image pixels that fall within the predefined range of skin threshold as skin pixel and remove the non-skin pixels which is out of the range. (OpenCV dev team, 2014) (Doxygen, 2017)

Figure 1.2-F5 Hand Region after Thresholding (Definition, 2013)

Morphological Transformations

The threshold image should be smoothened and filtered in order to reduce the noise that associated with it which will affect the performance on object detection and recognition. OpenCV library has also provide some morphological transformation functions for image optimization such as erosion, dilation and opening which will be used in this project (OpenCV dev team, 2014).

Figure 1.2-F6 Threshold image after perform Erosion and Dilation (OpenCV dev team, 2014)

Figure 1.2-F7 Reducing noise using Opening (OpenCV dev team, 2014)

Contour Detection

Contour refers to the outline or boundary of an object that form by connect all the continuous contour points by a curve. In image processing, it is a crucial element to be extracted from an image before perform object detection and recognition. It can be done by using cv2.findContour function from the OpenCV library and extract the maximum contour from the image which is our hand (Doxygen, 2015). Lastly the contour approximation is performed to approximate the contour shape for smoothen the contour edges.

Features Extraction

Features extraction refers to the process of transforming the input data which is the image into a set of measures for analysing and determine the meaning of the input data to perform specific function as an output. For this project, the features that require to be extracted from the image including the palm centre, convex hull, fingertips, hand defect points, area of hand, area ratio which is the percentage of area not covered by hand in convex hull and angle of finger as well. These features will be extracted by using set of algorithms and it will be further discuss in following chapter.

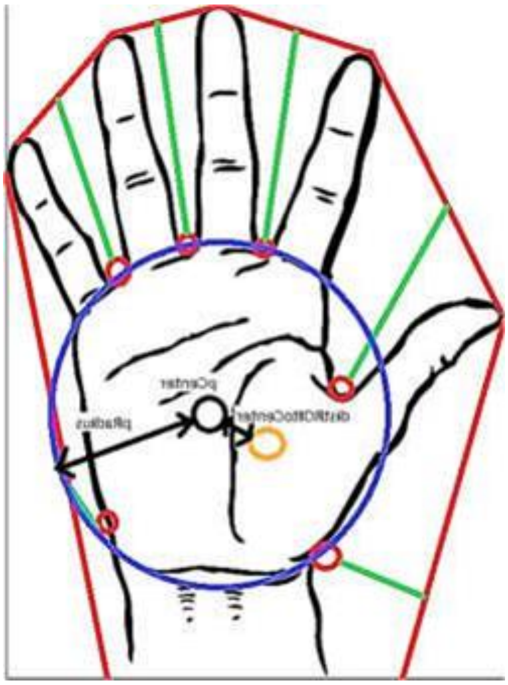


Figure 1.2-F8 Features Extraction (Popov, 2013)

1.3 Project Scope

Initially, the project is planned to allow gesture-based interaction between human and machine by developing a gesture recognition system in ADAS perspective which able to perform hand tracking and gesture recognition from a predefined vocabulary database in real-time by using specific image processing techniques and machine learning algorithms based on image data captured by a camera sensor such as Kinect and ToF camera. According to the intercepted gesture, the system will generate corresponding instruction to control the vehicle infotainment system function such as answer or reject incoming phone call, audio system control, temperature control as well as the GPS navigation control.

Due to strict time allocation, limited resources and lack of knowledge in high-level Python programming language, artificial intelligence and image processing techniques and algorithms, it is definitely an inevitable action to limit the scope of the project in order to deliver the proposed system without any delay on the project schedule. The entire procedures is designed to maintain a low computational cost with minimal hardware requirement and optimized to efficiently execute the required task. Some part of the initial planning have been withdraw from the project after serious consideration on the project feasibility on current level of study and limited project duration.

Therefore, the ultimate scope of this project is to develop a real-time gesture recognition system prototype that able to perform hand tracking and recognize set of static hand gestures that represent some car infotainment function by using several image processing techniques and algorithms on the image data that captured by laptop webcam or external webcam that mount in a car. Basically, the gesture recognition system can be split into five phase which included image acquisition, background subtraction and hand segmentation, features extraction and gesture recognition. Eventually, the system should be able to recognize 8 static hand gestures based on certain measure that take into account such as hand area, area ratio, number of defect point, number of finger, angle of finger and discriminate recognition error as much as possible. Instead of directly control the car infotainment function which is out of current level of study, the recognition result will be displayed with only function name once the gesture is being recognized.

Lastly, evaluation plan will also be conducted right after the system is completed. The system testing will be carried out in the room environment for the early development of the system. Once the system is getting mature and stable, it will be tested again in another environment which is inside the vehicle for stimulating the real situation. Detailed information of evaluation plan will be explained in Chapter 4.

1.4 Project Objective

To be accordance with the objective of ADAS which is to make cars safer to drive and assist driver in the driving process by enhance/ automate the vehicle system in order to reduce the rate of accident that cause by driver negligent, the project is being launched with the main objective of simplify and enhance the interaction between human and computer (HCI) by developing a low complexity real-time solution that enable hand gesture interaction between human and vehicle. More specifically, the project's objectives can be divided into the following sub-objectives:

i. To analyse the root cause of the increasing road accident rate and propose a solution to simplify and enhance the interaction between driver and vehicle without distract themselves to physically interact with the car infotainment system.

ii. To develop a real-time gesture recognition system that able to track and recognize set of human static hand gestures by implementing several image processing techniques and algorithms on the image data that captured by the webcam using Python programming language with OpenCV library.

iii. To evaluate the average recognition rate, accuracy and misclassification rate of the system throughout system testing which includes black-box testing, system performance testing and classification performance testing.

1.5 Impact, Significance and Contribution

The project initially aimed to allow gesture-based interaction between human and machine by developing a gesture recognition system for ADAS that able to perform hand tracking and gesture recognition in real-time by using specific image processing techniques and algorithms then generate instruction to control the vehicle infotainment system function based on the hand gesture recognized by the system. It will definitely provide a safety, comfort and convenience driving experience to drivers in controlling the vehicle infotainment system functions without distract themselves from driving to manually interact with the system using physical button, controller and touch screen interface.

However, the scope of the project has been limited by using a low complexity real time image processing techniques and algorithms to perform hand tracking and gesture recognition and using a real environment stimulation approach to display and represent system result visually due to several deficiencies and shortcomings that mention earlier in project scope such as time and resource constraints. Therefore, this project are contribute into the area of real time image processing, gesture recognition and visual representation of system functions in a stimulated environment. Besides that, the techniques and algorithms used are less complex and easier to understand compare to those computational intensive method that can be found in the journal article. So it can be useful for the people that don’t have strong image processing and python programming background to gain some basic knowledge and idea on the proposed topic. Other than that, the proposed image processing techniques, algorithms and codes can be used as a base for the future researcher or developer in building a more complex similar system and it can save a lot of time and effort.

CHAPTER 2: LITERATURE REVIEWS

2.1 Literature Review

This chapter will mainly focus on reviewing the journal article of some gesture recognition systems and algorithms that had been developed and introduced previously by researches and developers.

2.1.1 A Multisensor Technique for Gesture Recognition through Intelligent Skeletal Pose Analysis

In the previous work, RGB/ RGB-D camera such as Microsoft Kinect and Time-of-Flight (T-o-F) is used on the markerless CV hand tracking to analyse static and dynamic hand gesture in real-time using the raw colour and depth data to extract hand features such as hand and finger position from an estimated hand pose. Yet, this approach also present challenges for real-time gesture recognition due to frequent occlusion when the palm is not directly facing to the camera or the fingers blocked by another part of hand. The accuracy of the gesture interpretation can be disrupted and lead to unintended computer operations.

In the journal paper of "A Multisensor Technique for Gesture Recognition through Intelligent Skeletal Pose Analysis" (Rossol, et al., 2016) proposed a novel multisensor technique which aimed to improves the accuracy of hand pose estimation during real-time computer vision gesture recognition. This technique addresses the occlusion issue by placing multiple sensors at different viewing angle in performing pose estimation. Besides, they also built an offline model from an appropriately design subset of skeletal pose estimation parameter then used in real-time to intelligently select pose estimation. The experiment result shows a significant reduction in pose estimation error which is 31.5% compared to using only a single sensor and it can eliminate the false hand poses that interfere with accurate gesture recognition.

2.1.2 Contour Model-based Hand Gesture Recognition Using the Kinect Sensor

In order to cope with the main challenges in developing hand gesture-based systems which include locating the naked hand and reconstruct the hand pose from raw data that captured by using the Kinect sensor in the process of hand tracking, hand pose estimation and gesture recognition.

The journal paper "Contour Model-based Hand Gesture Recognition Using the Kinect Sensor" (yao & Fu, 2014) had proposed a novel procedure for capturing hand motion which is the semiautomatic labelling procedure with 14-patch hand partition scheme to reduce the workload of establishing sets of real gesture data. The method is being integrated into a vision based hand gesture recognition framework for the development of desktop applications.

Another challenge is the way to represent the hand model that allows the hand gesture database to be acquired efficiently by corresponding indexing and searching strategies. In order to deal with this challenge, they had also proposed a hand contour model that generate from the classified pixels and coded into strings in order to simplify the process of gesture matching and reduce the computational complexity of gesture matching. This framework allows hand gesture tracking in 3D space and support complex interactions in real-time. Their experiment result shows that gesture matching in this way can speed up efficiently to satisfy the requirements of real-time gesture recognition.

2.1.3 Static and Dynamic Hand Gesture Recognition in Depth Data Using Dynamic Time Warping

Hand gestures forms a powerful modality of interhuman communication which is intuitive and convenient mean for HCI. The journal paper of "Static and Dynamic Hand Gesture Recognition in Depth Data Using Dynamic Time Warping" (Plouffe & Cretu, 2016) discusses about the of natural gesture user interface development in tracking and recognizing hand gestures based on depth data collected by a Kinect sensor in real-time. An assumption has been made on the user's hand is the nearest object in the camera scene has determined the first segment of hand with the corresponding interest space. Besides, an improved block search scheme is proposed to reduce the scanning time on identify the first pixel of the hand contour and a directional search algorithm to identify the entire hand contour starting from the first pixel. Besides that, the k-curvature algorithm is used on fingertips localization over the hand contour. Eventually, the dynamic time warping (DTW) algorithm is being used for the selection of candidate gesture and compared the observed gesture with a series of pre-recorded reference gestures. The experiment result shows that an average result of 92.4% average recognition rate over sets of static and dynamic gestures.

2.1.4 Robust Fingertip Detection in a Complex Environment

Although CV technology has been developed rapidly, visual-based fingertips detection still presenting challenges as such detecting a flexible object with high level of freedom is difficult and nearly impossible to match all the finger shapes with a fixed template. Therefore, in the journal paper of "Robust Fingertip Detection in a Complex Environment" (Wu & Kang, 2016) proposed a robust fingertip detection algorithm that able to detect fingers in complex environment without requires any special devices. In hand region segmentation, dense optical flow region is being extracted and construct a skin filter with narrow ribbon to reduce the impact of clutter background and other region of skin colour. A novel block-based hand appearance model is being set up to assist hand and finger recognition. Lastly, a centroid circle method is also proposed for fingertips detection by looking for the local maximum distance outside the extended centroid distance circle. They believe that their algorithms will gives a good foundation for gesture recognition. Yet, the proposed algorithms still present some deficiencies.

2.1.5 Development of Gesture-based Human Computer Interaction Application by Fusion of Depth and Colour Video Stream

The journal paper "Development of Gesture-based Human Computer Interaction Application by Fusion of Depth and Colour Video Stream" (Dondi, et al., 2014) has presented a novel real-time gesture recognition system for the development of HCI application that exploits on using both depth and colour data. This system is using a ToF camera, MESA SR3000 that able to supply two kind of images per frame simultaneously which is a distance map and an amplitude map. An interesting part of this paper is all the gesture recognition process are only based on geometrical and colour constraints as the learning phase is not necessary. Even this method doesn't promise a higher precision but there will be a significant reduce on the computational time of recognition process and it is independent from the training set. Besides that, Kalman filter is implemented in this system in tracking hand and allow a precise recognition of hand in all frames. The entire procedure is designed to maintain a low computational cost and optimised to execute HCI task efficiently.

2.1.6 Gesture Interaction with Video: From Algorithms to User Evaluation

The journal paper "Gesture Interaction with Video: From Algorithms to User Evaluation" (Marilly, et al., 2013) proposed a vision-based approach that enabling natural HCI between user and a video meeting system in real-time either using static or dynamic gestures. The recognition process is split into two main functionalities which is the hand posture recognition and hand gesture recognition. The hand posture recognition consist of four steps which include skin segmentation, background subtraction, region combination, features extraction and classification. While the hand gesture recognition involving two steps which is tracking and recognition. Furthermore, this approach has enabled the combination of a signal similarity study with a data mining tool for dynamic gesture recognition. Last but not least, the paper is focus on the experimentation and user evaluation in order to reach a greater improvement, consider on user feedback and analysing performance in different environments for different users.

CHAPTER 3: SYSTEM DESIGN

This section is mainly focus on describing the overall project that has been designed, block diagram and some UML diagrams will be provided in order to give a clear picture on what the system will perform, how the system implement, what is the input/ output and so on. The UML diagrams that being used in this project are including the use case diagram and activity diagram. Thus, the proposed system can be more easily to understand by the readers.

3.2 Block Diagram

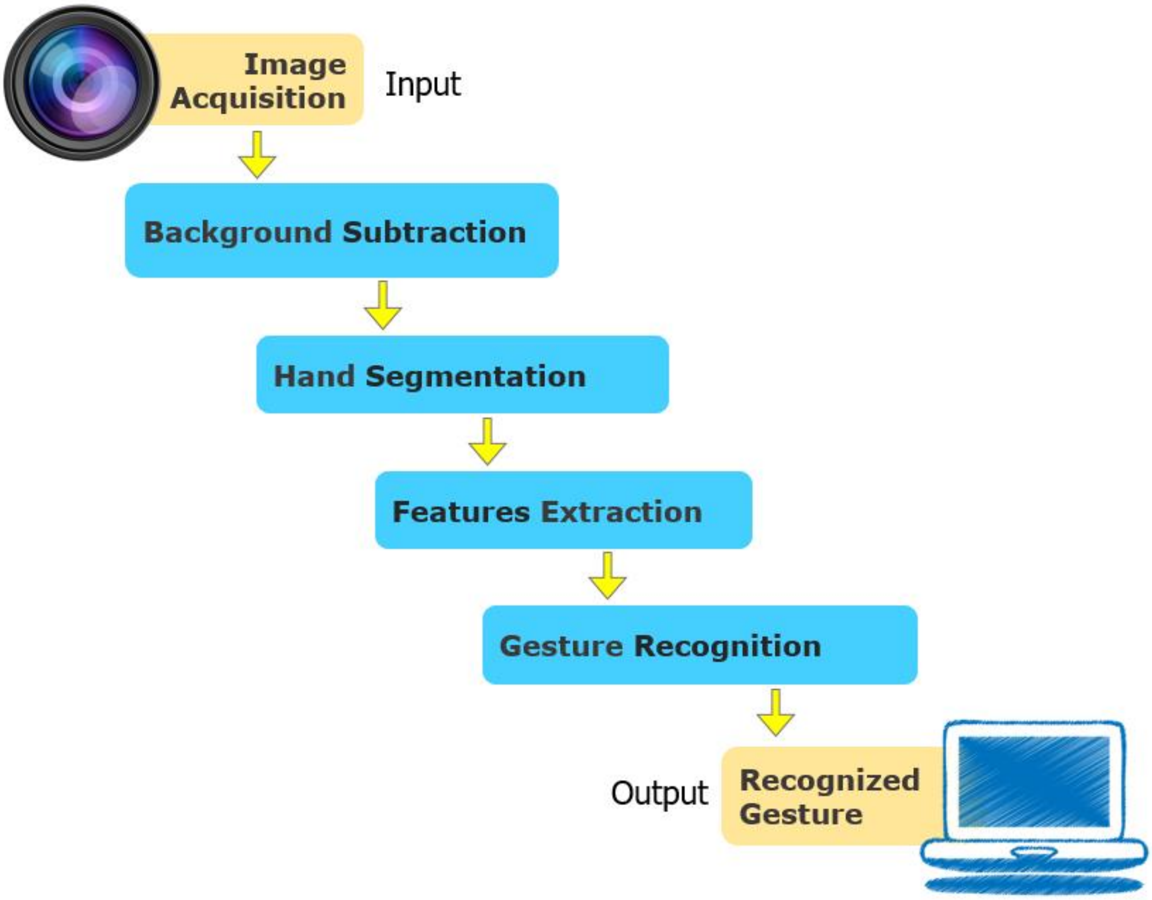


Figure 3.2-F1 Block Diagram for Real-Time Gesture Recognition System

Figure 3.2.1-F1 shows the overall system design of the real-time gesture recognition system. The system can be separate into five stages which is Image acquisition, background subtraction, hand segmentation, features extraction and lastly gesture recognition. Each stage must be completed before proceed to next stage. To make this clear, the image acquisition is the stage that capture user's hand as an input and prepare it for the next stage. The next four stages is the core processing stage of the system which include background subtraction, hand segmentation, features extraction and gesture recognition as well. The last part of the system will be the output stage that display the recognized result which representing the car infotainment function.

3.3 UML Diagrams

UML (Unified Modelling Language) diagram is useful visual representation of a software system design. It create a visual model of the software system and shows how the system actual implementation by including a set of graphic notation techniques. Furthermore, the present of UML diagrams in a developing object-oriented software system is important in specify, visualize, modify and document the system components. (TutorialPoint, n.d.) (SmartDraw, n.d.)

3.3.1 Use Case Diagram

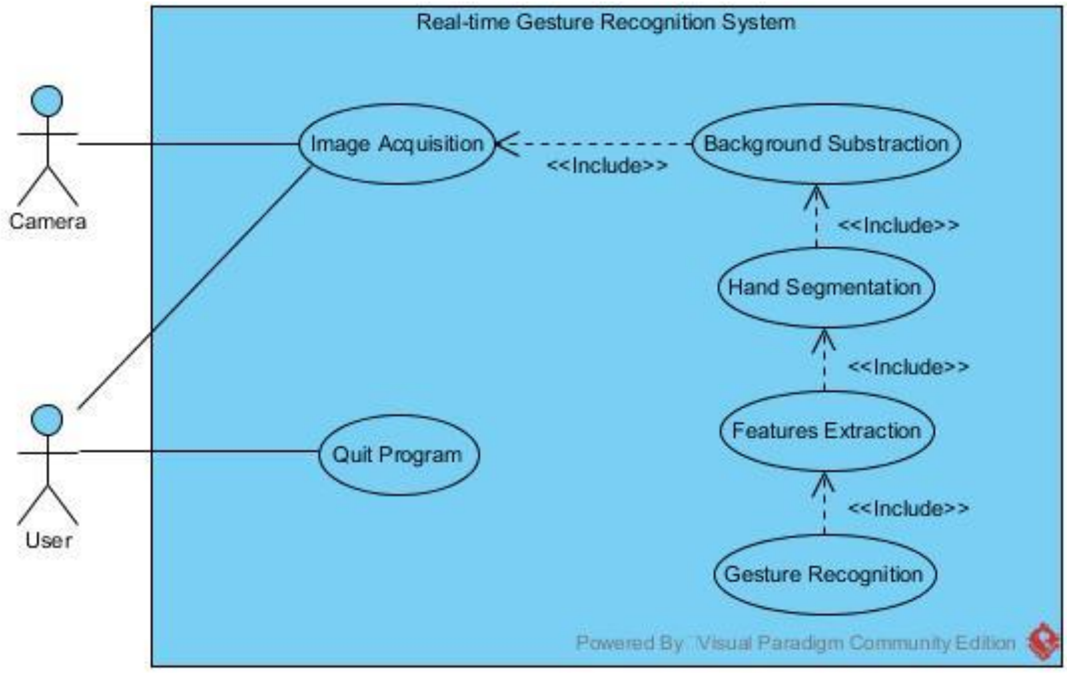


Figure 3.3.1-F1 Use Case Diagram for Real-Time Gesture Recognition System

Figure 3.3.1-F1 shows that the use case diagram has two actors which is the user and the camera. The user is the one that can initialize the system and quit the program after that. Besides, user also associate with the camera which is the laptop webcam or an external webcam that capture user image in real-time as the system input. In addition, the use case diagram also consist of 6 use cases where the image acquisition, background subtraction, hand segmentation, features extraction and gesture recognition are the core processing stages of the real-time gesture recognition system. The use cases of the system as below is describing the actions that perform by the actors and what will be the expected outcome.

Use Case 1: Image Acquisition

Actor: Camera/ User

Goal: To capture the video sequence of user hand image as system input.

Overview: The laptop webcam or external webcam is used to capture the user’s hand image. After that, the image frame is being resized, flipped and determine the ROI for further process in extracting useful information.

Use Case 2: Background Subtraction

Actor: Camera/ User

Goal: To process the video sequence for extracting the user's hand and remove unnecessary background and noise that associate with it.

Overview: Background subtraction, colour space conversion, thresholding and morphological transformation will be performed in order to prepare the binary image of user hand without unnecessary object and noises from a clustered background for the next processing stage.

Use Case 3: Hand Segmentation

Actor: Camera/ User

Goal: To obtain the hand contour and maximum contour of hand.

Overview: Hand contour is obtained from the binary image and get the largest contour in the image for the next stage.

Use Case 4: Features Extraction

Actor: Camera/ User

Goal: To obtain a set of hand features as the useful information that will be used for analysing and determine the meaning of the gesture input to perform specific function.

Overview: It is the process to transform the image data into a set of hand features such as palm centre, convex hull, fingertips, hand defect points, area of hand, area ratio which is the percentage of area not covered by hand in convex hull and angle of finger as well.

Use Case 5: Gesture Recognition

Actor: Camera/ User

Goal: To apply set of rules on the extracted information to determine the meaning of the gesture input and display the gesture that has been recognized.

Overview: The meaning of the gesture input will be determined by set of rules which include hand area, area ratio, number of defect point, number of finger, and angle of finger as well. The meaning of gesture will be displayed after it is being recognized.

Use Case 6: Quit program

Actor: User

Goal: To quit the program after it is being initialized.

Overview: User press on the “q” key to quit the program.

3.3.2 Activity Diagrams

The activity diagrams shows the program flows in the system that comprise of initial node, final node, activities, decision, action and so forth.

i. Image Acquisition

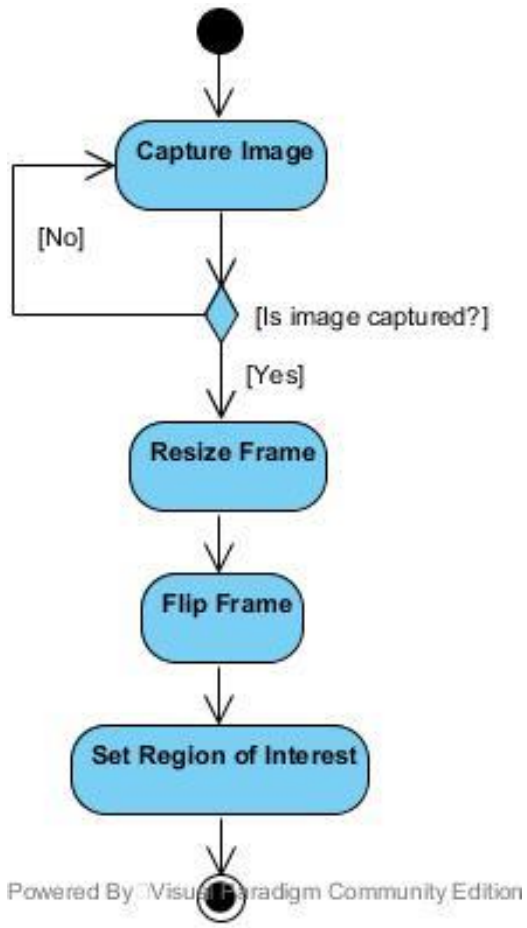


Figure 3.3.2-F1 Activity Diagram of Image Acquisition

In the beginning of the image acquisition, user's image will be captured by the laptop webcam or an external webcam. If the image is successfully captured, the image frame will be resized to a fixed width and flip the frame to avoid mirror view. Then, the recognizing zone which is the region of interest will be minimize instead of taking the overall video sequence.

ii. Background Subtraction

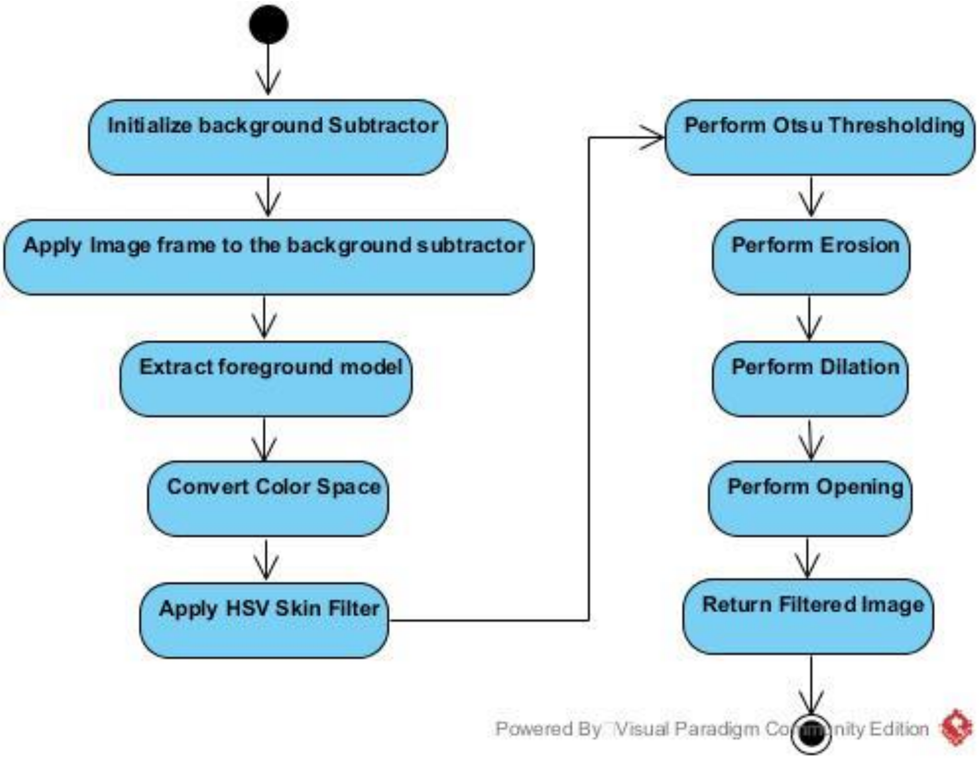


Figure 3.3.2-F2 Activity Diagram of Background Subtraction

In the background subtraction stage, the first thing to be performed is to initialize the background subtractor and apply the video sequence in order to extract the foreground model from the unnecessary background and noises. Next, the image need to be converted from the original RGB colour space into HSV colour space as mentioned in the previous chapter which is easier for hand detection and analysis. Then, skin filter will be applied to extract the image pixels that fall within the predefined range of skin threshold as skin pixel and remove the non-skin pixels which is out of the range. After that, Otsu thresholding is performed to transform the image into binary image which consist only black and white. The last step will be morphological transformation that consist of erosion, dilation and opening then return the filtered image.

iii. Hand Segmentation

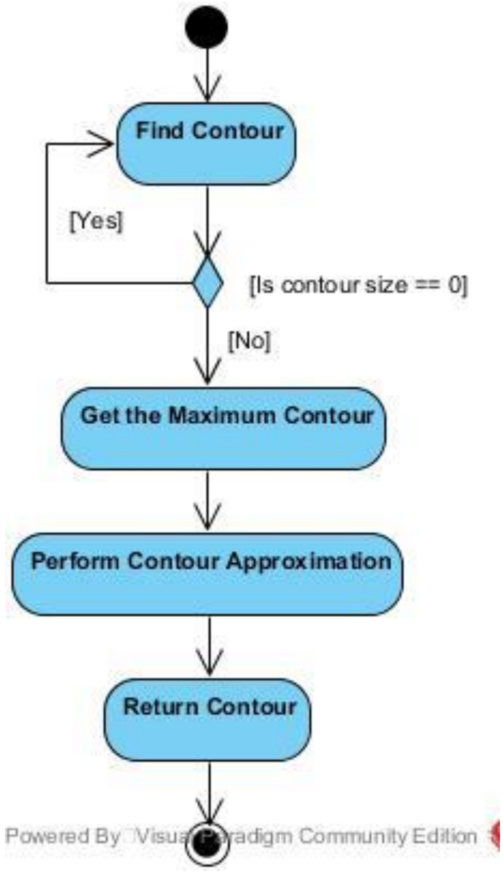


Figure 3.3.2-F3 Activity Diagram of Hand Segmentation

Figure 3.3.2-F3 shows the third stage which is the hand segmentation that perform contour detection which find the largest contour in the image. Then contour approximation has to be performed to approximate the contour shape for smoothen the contour edges. Lastly, the detected contour will be returned to main for further processing.

iv. Features Extraction

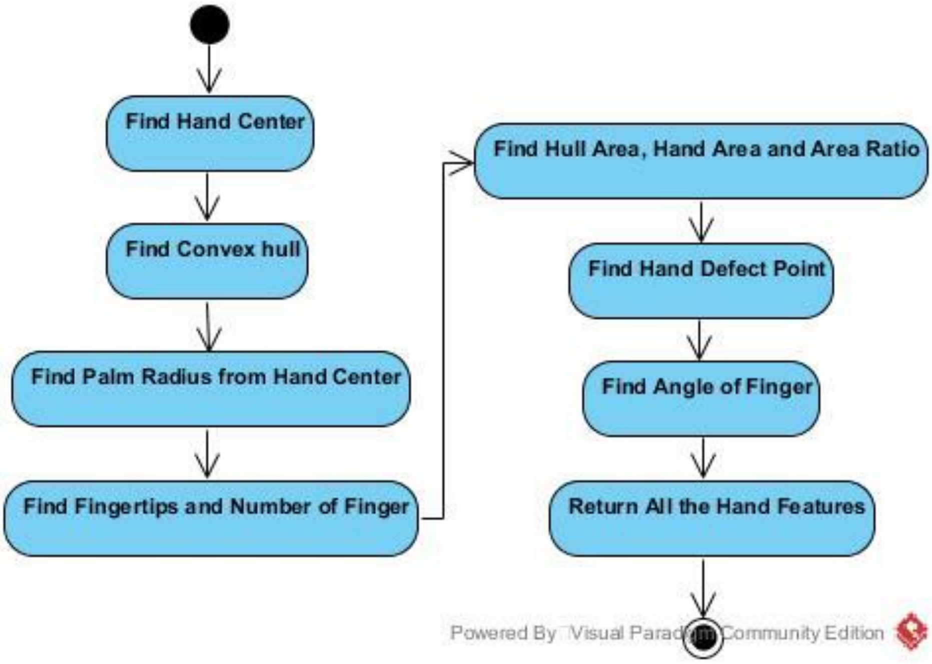


Figure 3.3.2-F4 Activity Diagram of Features Extraction

Figure 3.3.2-F4 shows the features extraction stage that firstly find the hand centre. Then, find the convex hull followed by get the palm radius from the centre of palm to the most extreme points in the convex hull. The next step will be looking for the fingertips location and number of finger. After that, the hull area and hand area will be used to calculate the area ratio. The number of hand defect point and the angle of finger also need to be calculated. Finally, the last step will be returning all the extracted hand features for gesture recognition.

v. Gesture Recognition

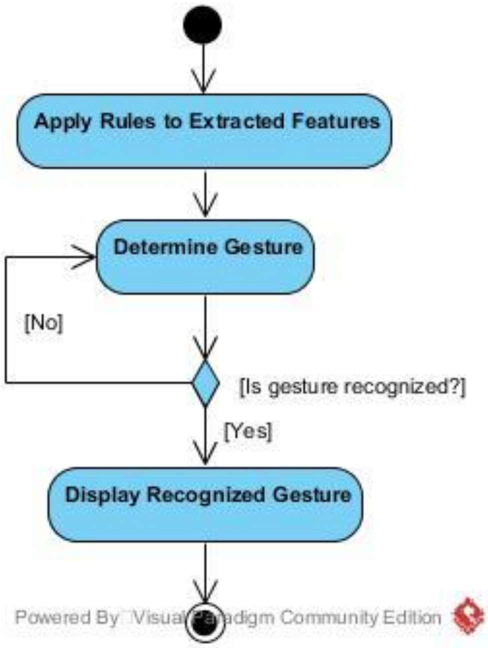


Figure 3.3.2-F5 Activity Diagram of Gesture Recognition

Figure 3.3.2-F5 shows the gesture recognition stage that applying set of rules to the extracted features and determine whether the gesture is being recognized. If yes, the recognized gesture will be display to indicate the recognition is successful. Else, the system will continue determining gesture.

vi. Quit Program

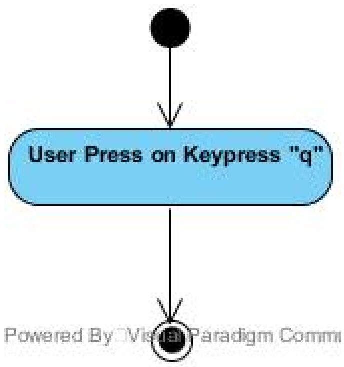


Figure 3.3.2-F6 Activity Diagram of Quit Program

The quit program function shows that user can press on ‘q’ to quit program.

CHAPTER 4: DESIGN SPECIFICATIONS

4.1 Methodology

Among various system development methodologies, the Evolutionary Prototyping which is one of the prototyping methodologies is being selected in developing this project. The basic idea of this methodology is develop an initial prototype and keeps on refining the system requirement through number of cycles until the final system is completed and satisfied by the client (Sommerville, 2000). In this case, the project supervisor and the developer himself will be the client that responsible to evaluate and provide feedback based on the prototype created. This methodology can be separated into four phases, which is initial concept, design and implementation of initial prototype, prototype refinement until it is acceptable and lastly deliver of the complete system.

The reason why Evolutionary Prototyping methodology is chosen is because this project is to develop a software system that requires continuous feedbacks and suggestions from the client to improve the prototype until the final system is completed and delivered. Besides that, Evolutionary Prototyping can speed up the system development process and improve the quality of the final system since it requires going through several prototypes until the final version of system is fulfil the predefined requirements and functionalities. It will also help in increase the satisfaction level of the client since the prototype is generated based on the requirement specified by the end user.

However, there are also some drawback by using Evolutionary Prototyping which include higher rate of failure in develop the complete piece of system that satisfied all the requirements and functionalities due to lack of planning effort in this system development methodology. The prototypes are just created based on the initial concept without further planning and analysing on the project feasibility and the money, time and effort that sacrificed previously will be wasted if the project failed. Furthermore, the completion date and project cost is difficult to be determined because the system requirements may be change from time to time based on the client.

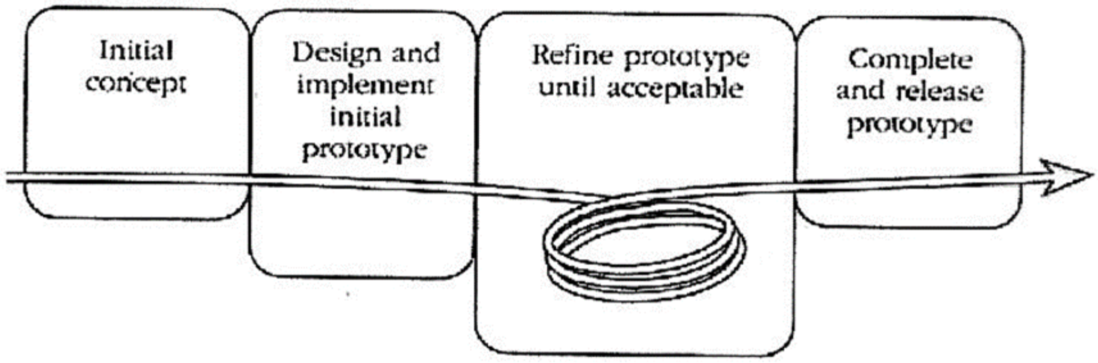


Figure 4.1-F1 Evolutionary Prototyping Model (Weinberg, n.d.)

Initial concept

This is the phase that the initial idea of the proposed system is created and begin to gather the related information from the existing literature such as journal article and website. The basic requirements of the system are being analysed by researching the literature on the general image processing techniques, algorithms and standard procedure to develop a real-time gesture recognition system. At the end of this phase, it should be able to come out with a project plan, list of initial requirements, and a list of required resources and the methodology that used to develop the system. Thus the initial concept of this project is to develop a real-time gesture recognition system prototype that able to perform hand tracking and gesture recognition on set of hand gestures that represent some car infotainment function.

Design and implement initial prototype

At this phase, all the information that previously gathered will be used as a references to design the actual implementation of the real-time gesture recognition system and determine which gesture recognition technique, image processing techniques and algorithms to be applied based on the listed requirements which related to the project objective sets. Some UML diagrams also will be designed in order to allow developer and client understand the design of the system in depth and provide a clear picture of how the system structure look like.

Besides that, it is necessary to determine a set of rules to be applied on the feature extracted for differentiating gesture performed and test whether the system is able to perform task based on the requirements. After all, the complete system design will be initially implemented and quickly come out with the first prototype that fulfil all the basic requirements of the project. The prototype will be tested and evaluated by the user in order to collect feedback and suggested improvement for the next prototype. At the end of this phase, the developer will need to come out with a list of validated requirements, system design, and evaluation and feedback from the users about the missing requirements and so forth.

Refine prototype until acceptable

This stage will mainly focus on refinement and modification of the system design through the observation and evaluation from the testing result and incorporate those modified requirements into the following prototype. Besides, the quality of the system in the following prototype also need to consider carefully because it will getting closer and closer to the final system. Therefore, the prototype will be refined again and again and test through experimentation until it met the project objective sets.

Complete and release prototype

In the last phase of Evolutionary Prototyping, a piece of complete and functioning real-time gesture recognition system will be fully developed based on the validated final requirements and deliver to the client as an approved system with the required functionality and quality that built with it. This is the phase that all the scope and objectives of the project will be fulfil and satisfied by the client. In the end of this phase, a piece of fully functioning real-time gesture recognition system that project report that contain all the information of the project will be generated and submitted.

4.2 Technology Involved

4.2.1 Software

i. **PyCharm Community**

Figure 4.2.1-F1 PyCharm Logo (Jetbrains, 2016)

PyCharm community is an open source version integrated development environment (IDE) from the Czech company JetBrains. Although it may be not necessary in developing a python programming-based project, but it is a great platform that offer various powerful features for improving productivity such as intelligent coding assistance which allow easy code navigation, error checking, quick fixes and refactoring as well. (JetBrains, n.d.)

ii. OpenCV 3.4.1

Figure 4.2.1-F2 OpenCV logo (Shavit, 2006)

OpenCV (Open Source Computer Vision) is an open source library of programming functions mainly aimed at real time computer vision that originally developed by Intel. The library is cross-platform and free for use under the open-source BSD license. OpenCV has provided various functions that related to object tracking and image processing that will be used in the development of implementable algorithms which meets the aim of this project.

4.2.2 Hardware

i. **ASUS TUF FX504GD Laptop**

Figure 4.2.2-F1 ASUS TUF FX504GD Laptop (Cuyugan, 2018)

The hardware that used to develop this system include an ASUS TUF FX504GD Laptop that equip with specifications as below:

- Processor: Intel® Core™ i5-8300H CPU @ 2.30GHz
- Installed memory (RAM): 12GB SDRAM
- Operating System: Window 10 Home Premium 64-bit Operating System.
- Graphic Card: NVIDIA GeForce GTX1050 4G DDR5 VRAM
- Storage: 1TB SSHD
- Camera: HD Web Camera

4.2.3 Programming Language

i. **Python 3.6.4**

Figure 4.2.3-F1 Python Logo (www.python.org, 2008)

The program will be written in high-level Python programming language which provides easy syntax that allow quick coding in fewer steps to complete certain statement and function compared to Java or C++. Besides, Python also provides various standard libraries which enable the execution of complex functionalities easily.

4.3 Functional Requirements

i. The laptop webcam and an external webcam are able to capture the video sequence in real-time.
ii. The system able to produce multiple frame to display the captured video sequence and the processed image.
iii. The system able to detect the skin region from the captured video sequence.
iv. The system able to detect user's hand contour from the segmented image.
v. The system able to extract the hand features such as hand contour, hand centre, hand radius, convex hull, convexity defect points and fingertips.
vi. The system able to display the extracted hand features.
vii. The system able to display the recognized result.

4.4 Assumptions

There are several assumptions that have been made throughout the system design in order to avoid error and undesirable output which include:

i. User is expected to include only one hand the in the camera scene.
ii. User's left hand is expected to be the only active object in the camera scene.
iii. User is expected to wear a long sleeve shirt that not close to our skin colour.
iv. User's hand is expected to be naked without any accessory or jewellery.
v. There must be sufficient lightning in the operating environment.
vi. Clustered background has to be avoided.

4.5 System Performance Definition

In the development of a system, there is a need to specify the system performance definition which is the predefined standard of measure in evaluating the functionalities and the performance of system in order to achieve better improvement. Therefore, the system performance definition that being used in this real-time gesture recognition system includes **system functionalities** which will be evaluated through black-box test and determine whether the system is fulfils all the functional requirements.

Besides that, the system performance definitions also include system performance which measure by the **average recognition rate** in recognizing set gestures throughout few iterations and determine the recognition rate of the system in successfully recognizes a gesture throughout all the iterations.

Another performance definition is the **classification performance** of the system which measure by the accuracy and misclassification rate using the confusion matrix theory. This is to determine whether the result of recognition is true positive which mean the recognition result is correct or false positive which representing the gesture perceived is wrongly recognized as another gesture. (Data School, 2014)

4.6 Evaluation Plan

The evaluation plan in the project development is crucial for evaluating the system performance that have mentioned in the system performance definition and determine whether the system is satisfies the specified requirements and project objective. For the functional testing, black-box testing will be conducted on evaluating several test cases which is the functional requirements that have stated earlier. The test case is passed if the actual result of the system outcome is fulfil the expected result, else it is failed.

For the non-functional testing which is the performance testing that includes the system performance testing and classification performance testing, the evaluation will be separated into two part which is carry out in room environment and car environment. In each environment, all the predetermined gesture will be tested in several iteration to increase the reliability of the test result.

To find out the system performance, the recognition result in each iteration will be recorded in whether the gesture is successfully being recognized or not being recognized. There will be a fixed period for each iteration which is 3 seconds. If the gesture is being recognized in 3 seconds, then it will be marked as a successful recognition. Else it will be marked as unsuccessful recognition. After that, the average of the overall result will be taken to determine the average recognition rate.

Lastly, the classification performance will be evaluated through the accuracy and misclassification rate from the overall classification result. Each iteration is done when the first classification result is shown. If the classification result is match with the testing gesture, then the result will be recorded as true positive. Else if the classification result is different with the testing gesture, it will be recorded as false negative. After all iteration is completed, the classification result will be used to calculate the accuracy and misclassification rate.

4.7 Project Timeline

The Gantt chart is provided in this section to show the timeline and planning for the three stage of FYP which is the IIPSPW, FYP 1 and FYP 2.

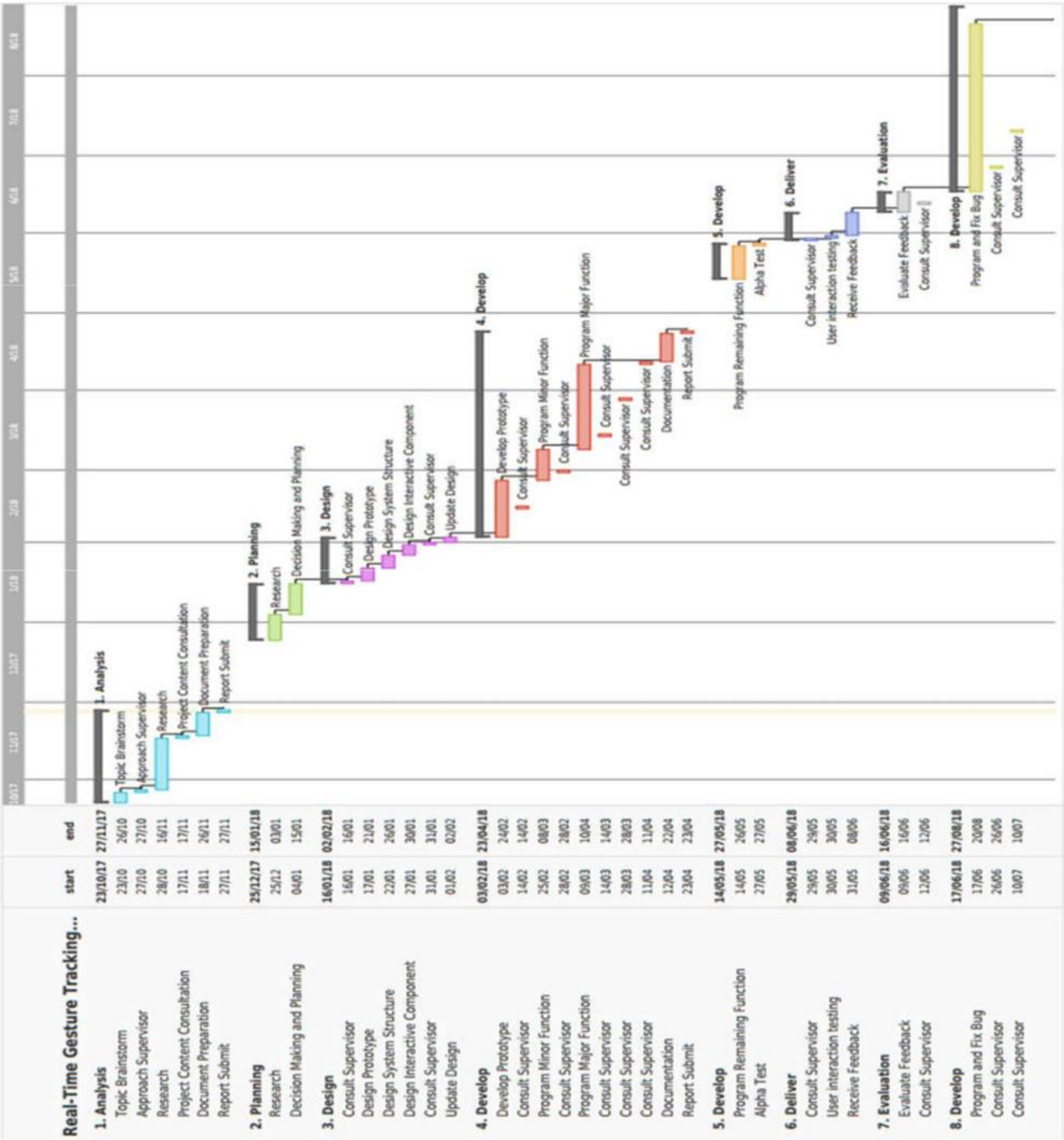


Figure 4.7-F1 Gantt Chart

CHAPTER 5: IMPLEMENTATION & TESTING

5.1 System Implementation

This chapter will mainly focus on the implementation stage of the project which describe the use of information from the system design and design specifications in developing and implementing a set of required algorithms and calculation to achieve the project objectives. Various OpenCV functions that relate to object tracking and recognition will be used to assist the development of the system. Basically, the real-time gesture recognition system can be split into four stages which included image acquisition, background subtraction and hand segmentation, features extraction and gesture recognition.

5.1.1 Image Acquisition

In the main function of the program, the video capturing function which is a reference to the default webcam instance is being taken by using *cv2.VideoCapture(0).* The function parameter need to be changed to another device registration number if another external webcam is used as the system input to capture user image.

If the video sequence is successfully captured, the video frame will be resized to a fixed width and height using *imutils* library then flip the frame to avoid mirror view. Next, the recognition zone in the overall video sequence will be minimized into a preferable size by setting the region of interest in order to improve the system efficiency in looking for the hand region.

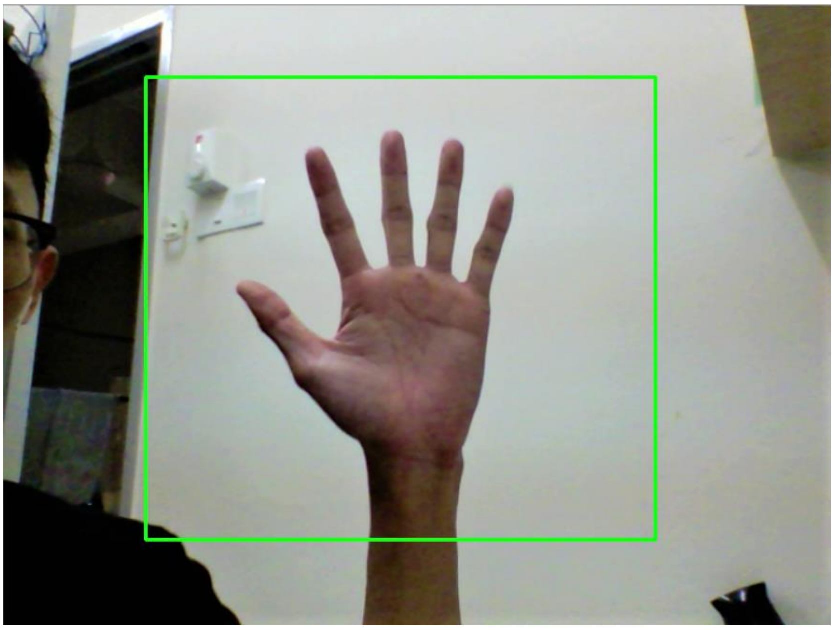


Figure 5.1.1-F1 ROI in Overall Video Sequence

Figure 5.1.1-F1 shows the green rectangular which is the region of interest in the overall video sequence that will be used for the next processing stage.

5.1.2 Background Subtraction

The first thing to be performed in the background subtraction stage is initializing the background subtractor by using *cv2.createBackgroundSubtractorMOG2()* with the threshold value of 150 and set the shadow detection to false to avoid the shadow being recognized as part of the object. The next step will be applying the video sequence to the background subtractor that has been initialized with the learning rate of 0 to extract the foreground model from the unwanted background and noises.

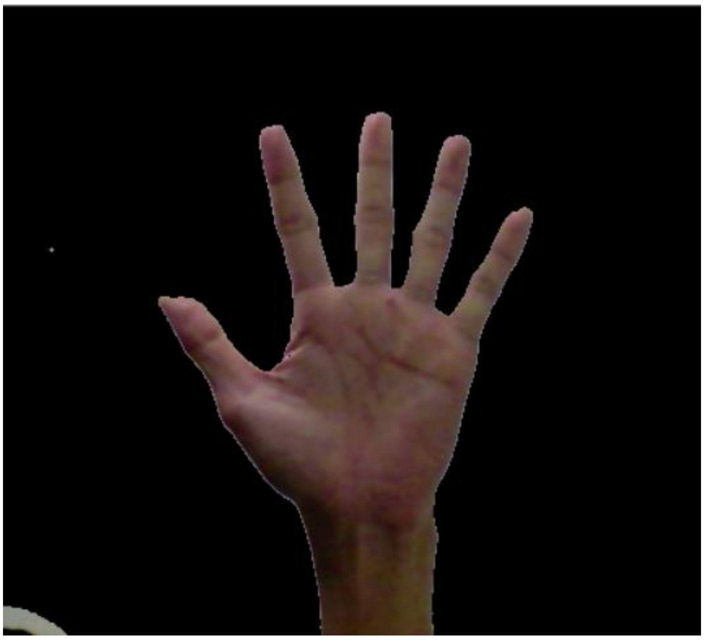


Figure 5.1.2-F1 Extracted Foreground Model

Since the original video sequence are in RGB colour space which is not suitable for colour-based detection and analysis. So it has to be converted into HSV colour space as mentioned in previous chapter. Then, the HSV skin filter *cv2.inRange()* will be applied to the image in extracting the image pixels that fall within the predefined range of skin threshold as skin pixel and remove the non-skin pixels which is out of the range. Before that, the range of skin threshold of the HSV skin filter must be determined in advance by creating a new Python file that mainly for adjusting on the range of colour value by using real-time track bar.

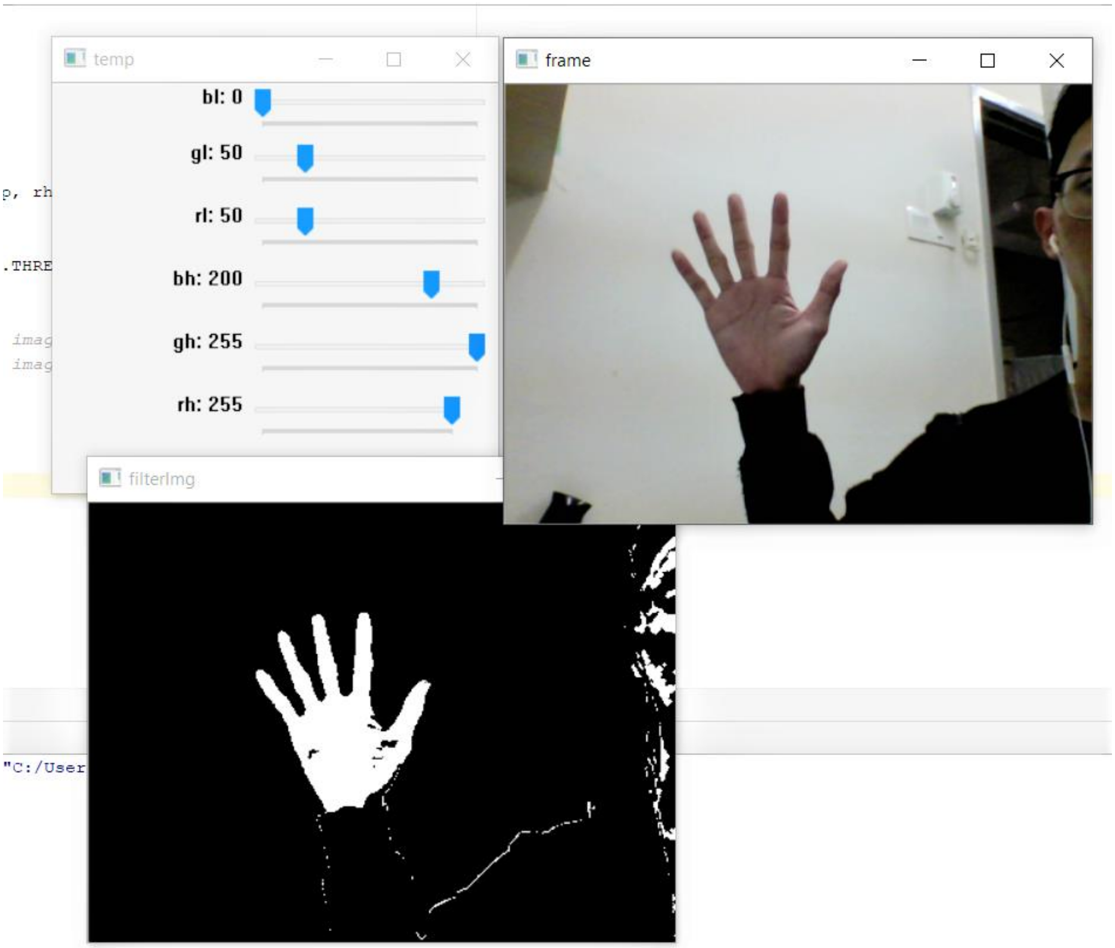


Figure 5.1.2-F2 Testing for Acquiring Range of Skin Threshold

After that, Otsu thresholding will be performed by using *cv2.threshold()* with an extra flag *cv2.THRESH_OTSU* to transform the image into binary image that only consist of black and white colour based on the pixel value to the threshold value. The detected hand region will be set to white and other unwanted region will be set to black.

The last step of the background subtraction that will be performed is the morphological transformation which consist of erosion, dilation and opening to smoothen the threshold image and reduce the noise that associate with it. It can be achieved by using *cv2.erode(), cv2.dialate()* and *cv2.morphologyEx()* with extra flag of *cv2.MORPH_OPEN* then return the filtered image back to the main for next processing stage.

Figure 5.1.2-F3 Filtered Image

5.1.3 Hand Segmentation

In the hand segmentation stage, the filtered image from the previous stage which consist only the hand region will be applied to the find contour function by using *cv2.findContours()* and get the maximum contour in the image based on the contour area. Then, the contour approximation is performed to approximate the contour shape and smoothening the contour edges by using *cv2.approxPolyDP()* with an approximated curve for epsilon. Eventually, the hand contour will be drawn by using *cv2.drawContours()* and return to main for further processing.

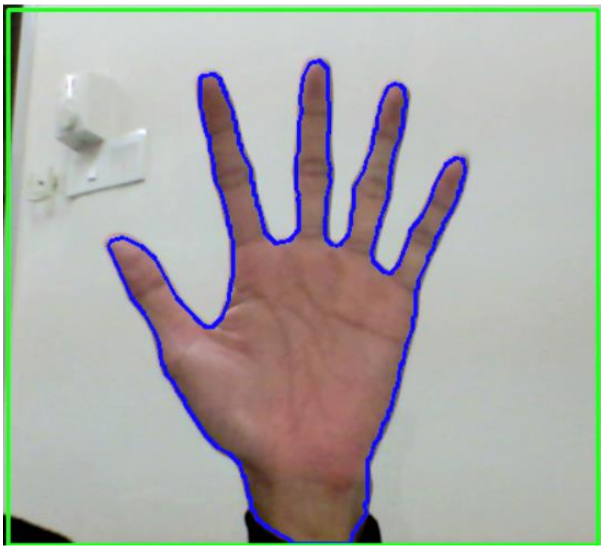


Figure 5.1.3-F1 Extracted Hand Contour

5.1.4 Features Extraction

Features extraction is one of the core processing stage in the real-time gesture recognition system that transform the segmented image into a set of measures which will be used for analysing and determining the meaning of gesture in the gesture recognition stage. Firstly, the centre of the hand can be found by using the moments of the contour with the function *cv2.moments()* and calculate the centre mass of the hand contour with following formula:

$$(cx,\ cy) = (\frac{M_{10}}{M_{00}},\ \frac{M_{01}}{M_{00}})$$

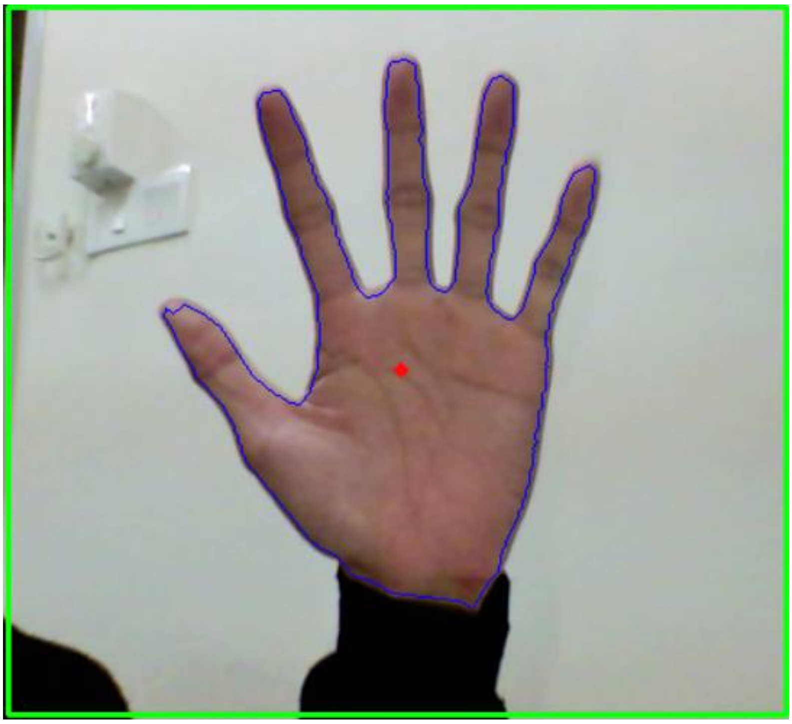


Figure 5.1.4-F1 Centre Mass of Hand

Next, the convex hull can be found by using *cv2.convexHull* in order to get the palm radius by calculating the maximum Euclidean distance between the centre of palm and the most extreme points in the convex hull. Scikit-learn has provides a function *pairwise.euclidean_distances()* to find the distance between one point to multiple points. Then the radius of palm will be assumed as 40% to the maximum distance from the hand centre.

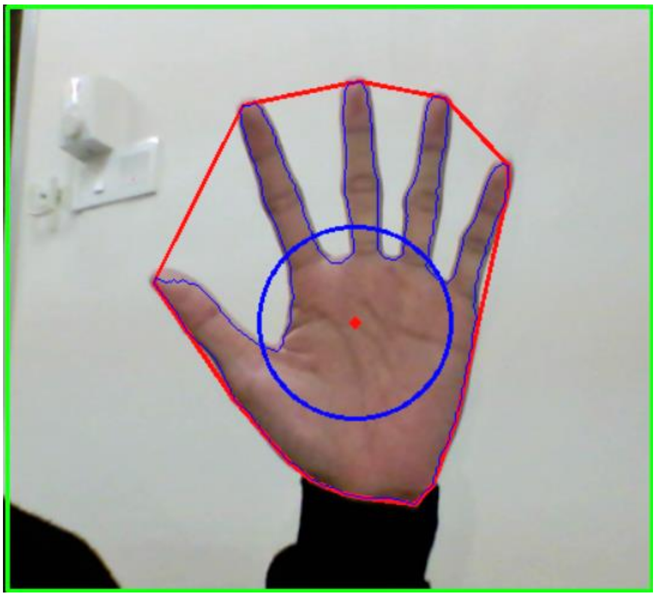


Figure 5.1.4-F2 Convex Hull and Radius

Besides that, the position of fingertips can be found by applying several steps to the hand centre, palm radius and the convex hull points that have been extracted. The first step will be eliminating the convex hull points that are very close to each other. Secondly, the convex hull points that are too near or too far to the centre of hand have to be eliminated with the minimum and maximum finger length threshold to ensure only the finger part is detected. Lastly, the result of fingertips detection has to be optimized so that there will be not more than 5 fingers.

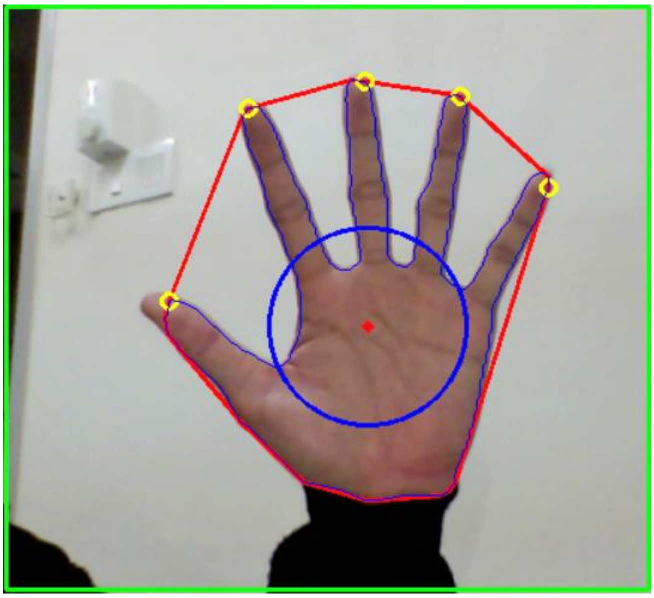


Figure 5.1.4-F3 Fingertips Detection

In addition, the hull area and hand area also need to be acquired by using cv2.contourArea in order to calculate area ratio which is the percentage of area that are not covered by hand in the convex hull by following formula:

$$Area\ Ratio = \frac{Hull\ Area - Hand\ Area}{Hand\ Area} \times 100$$

Other than that, the convexity defects is the cavity in the convex hull that formed when there are two or more hull point in the convex hull which is the fingers . It can be found by using function *cv2.convexityDefects()* and return with four values which include the start point, end point, farthest point and the approximate distance to the farthest point. In this case, only the first three values will be used in calculating the convexity defects.

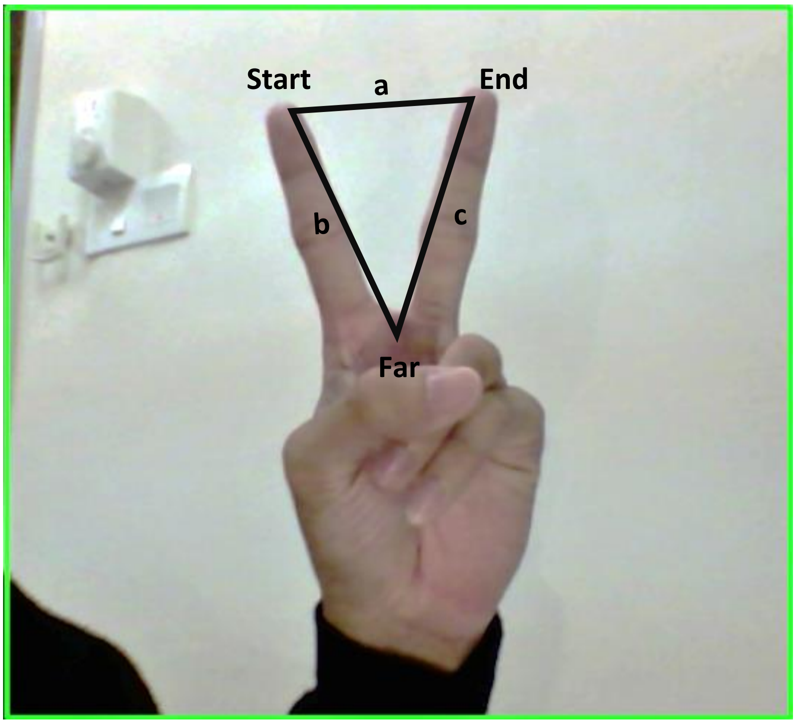


Figure 5.1.4-F4 Start, End and Farthest Point in the Convexity Defect

After that, the length between the three points will be represented by a, b, c and it have to be calculated by using the distance formula as below:

$$a = \sqrt{(start[0] - end[0])^2 + (start[1] - end[1])^2}$$

$$b = \sqrt{(start[0] - far[0])^2 + (start[1] - far[1])^2}$$

$$c = \sqrt{(end[0] - far[0])^2 + (end[1] - far[1])^2}$$

Once the length between the three points have been found, the angle between the two fingers can be determined by using the Cosine rule:

$$a^2 = b^2 + c^2 - 2bc \cos A$$

$$A = \cos^{-1}(b^2 + c^2 - a^2 / 2bc)$$

The distance between the convexity defects and convex hull also being taken as a measure to determine the number of convexity points to eliminate the defect points that are too close to the convex hull. It can be achieved by following formula:

$$s = (a + b + c)/2$$

$$ar = \sqrt{s \times (s - a) \times (s - b) \times (s - c)}$$

$$d = 2ar/a$$

The last step is to take in the result from two previous step as the parameter to determine whether it is a convexity defect points between two fingers. If the results is fulfil the following statement, it is consider as a defects point.

$$If \left| \begin{array}{c} Angle\ between\ two\ fingers\ \leq 90 \\ Distance\ between\ the\ convexity\ defects\ and\ convex\ hull > 45 \end{array} \right|$$

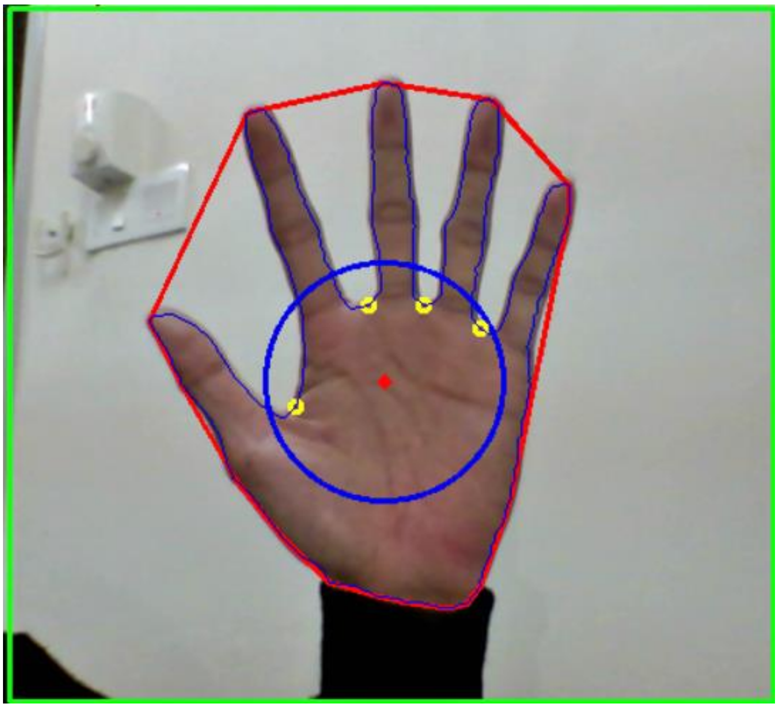


Figure 5.1.4-F5 Convexity Defect Points of Hand

The last features to be extracted is the angle of finger which in either involving one finger or two fingers with different formula. If there is only one finger, the angle of finger will be determined based on the hand centre coordinate (cx, cy) and the fingertips coordinate (x, y) by using following formula:

$$A = \tan^{-1}(cy - y)/(cx - x)$$

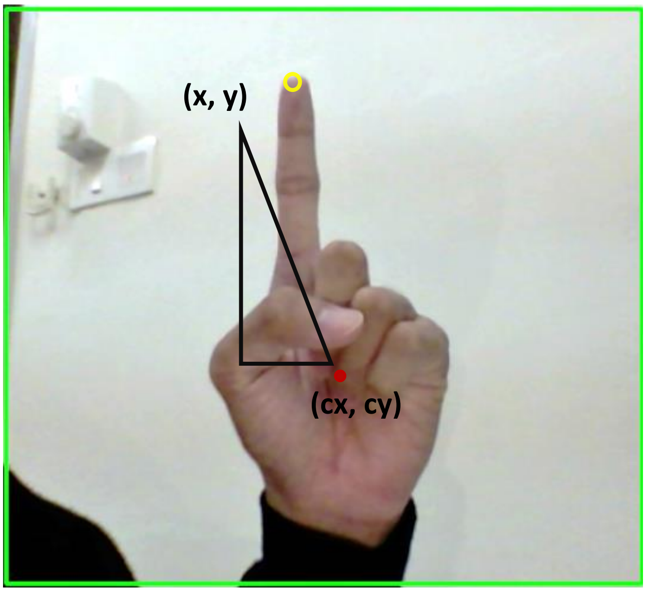


Figure 5.1.4-F6 Calculate the Angle of One Finger

Else if there are two finger, the angle of finger will be determined based on the two fingertips coordinate and the hand centre coordinate by using the cosine rule as mentioned as above.

Eventually, all the extracted features will be returned to the main as a set of measures that will be used for analysing and determining the meaning of gesture in the gesture recognition stage.

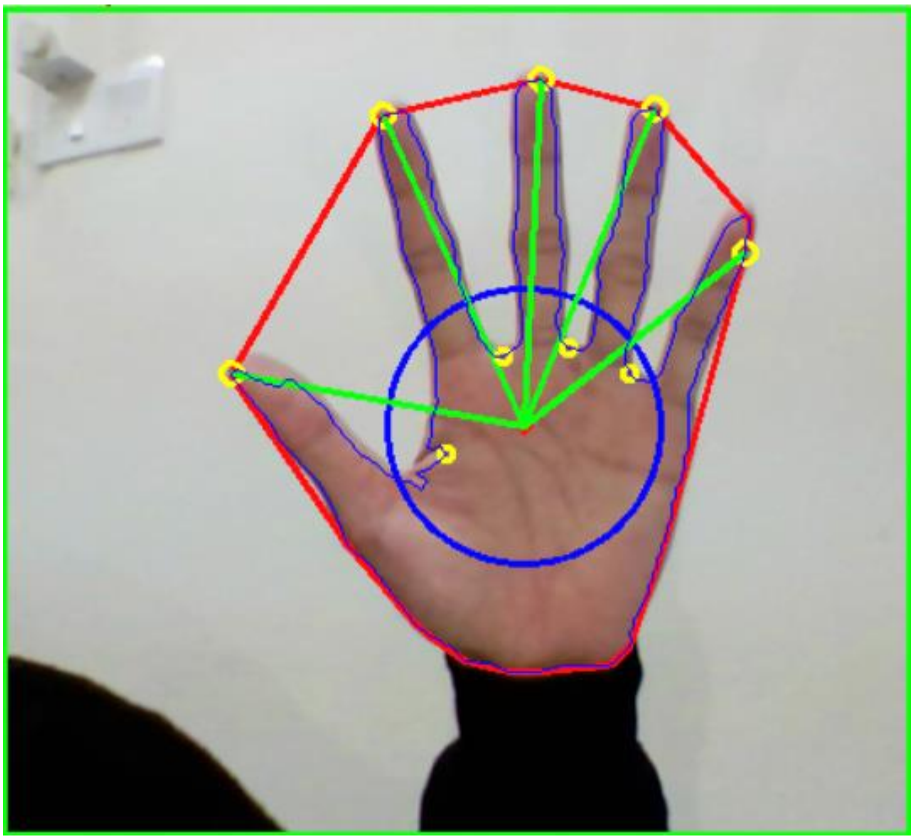
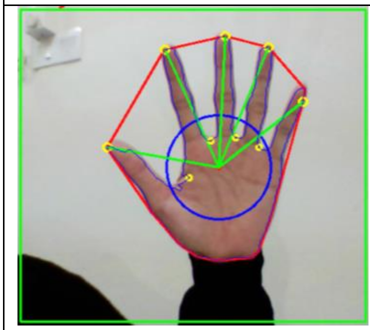
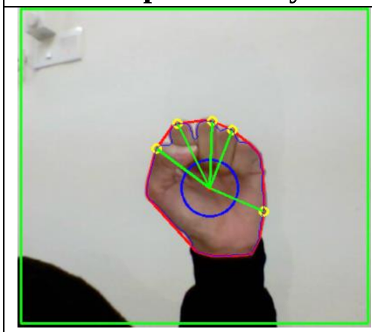
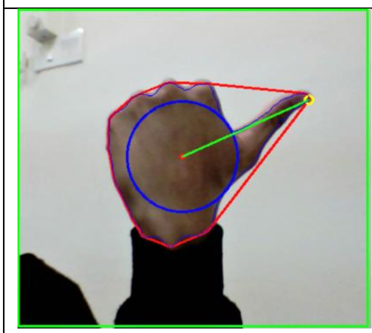
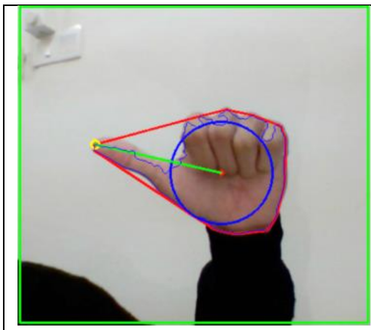
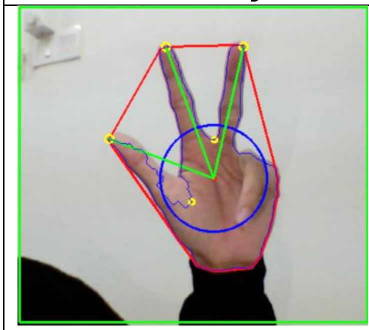
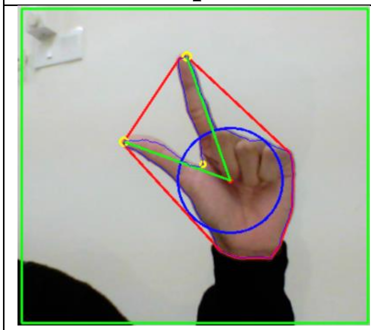


Figure 5.1.4-F7 Display of All Extracted Features

5.1.5 Gesture Recognition

Gesture recognition is the final stage of the real-time gesture recognition system that applying set of rules from the hand features that extracted from previous stage to build the gesture recognition model in order to determine gesture and discriminate recognition error as much as possible. There are 8 set of different rules for 8 different gestures which will be described as table below:

Gesture	**Rule of the gesture recognition model**					
	Defect Point	**Number of Finger**	**Hand Area**	**Area Ratio**	**Finger Angle**	**2 Finger Angle**
Hand is not in frame	-	-	<2000	-	-	-
 **Full palm – Play**	4	5	>2000	40 – 75	-	-
 **Punch – Pause**	0	0	>2000	4 – 15	-	-
 **Thumb right – Accept call**	0	1	>2000	15 – 30	0 – -50	-

 **Thumb left – Reject call**	0	1	>2000	15 – 30	0 – 50	-
 **Three finger – Volume up**	2	3	>2000	55 – 80	-	-
 **Thumb with one finger – Volume down**	1	2	>2000	28 – 70	-	50 – 90

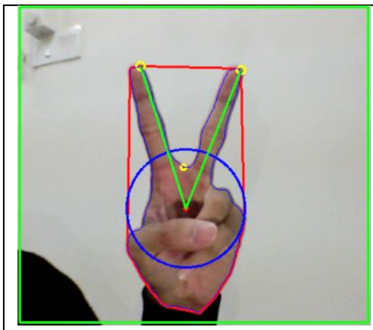
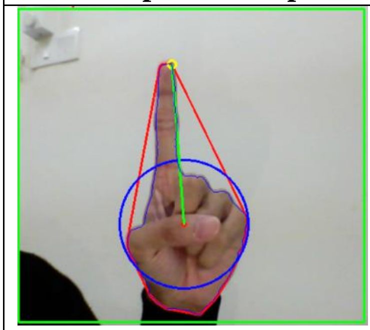
 **Two finger – Temperature up**	1	2	>2000	28 – 70	-	0 – 50
 **One finger – Temperature down**	0	1	>2000	30 – 50	70 – 90, -70 – -90	-

Table 5.1.5-T1 Gesture recognition model

5.2 System Testing

The system testing is the process to evaluate whether the system has fulfils the system requirements which is the expected functionalities that will be performed by the system and evaluate on the system performance by using appropriate standard. For this project, there will be functional testing which evaluate on the system functionality and the non-functional testing which will evaluate on the system performance in terms of average recognition rate and classification performance.

5.2.1 Black-Box Testing

Test	Test Case	Expected Outcome	Actual Outcome	Result
1	Initialize camera	Display the video feed in new window	Original video feed displayed in new window	Pass
2	Background subtraction	Display the hand without background	Hand is displayed with black background	Pass
3	Colour space conversion	Display HSV image in new window	HSV image is displayed in new window	Pass
4	Image filtering	Reduce noise in the image	Noise is reduced	Pass
5	Skin detection	Display hand region in white with black background	Hand region is displayed in white without background	Pass
6	Contour detection	Draw contour around hand region	Hand contour is drawn in blue colour along hand region	Pass
7	Features extraction	Display of hand centre	Hand centre is displayed in red dot	Pass
		Display of convex hull	Convex hull is drawn along the convex points	Pass
		Display of palm radius	Palm radius is drawn in blue circle	Pass
		Display of fingertips	Fingertips is displayed in yellow dots	Pass

		Display of defect points	Defect points is displayed in yellow dots	Pass
8	Gesture recognition	Display “Play” when show full palm gesture	“Play” is displayed	Pass
		Display “Pause” when show punch gesture	“Pause” is displayed with slightly delay	Pass
		Display “Accept Call” when show thumb right gesture	“Accept Call” is displayed	Pass
		Display “Decline Call” when show thumb left gesture	“Decline Call” is displayed	Pass
		Display “Volume Up” when show thumb and two finger gesture	“Volume Up” is displayed with slightly delay	Pass
		Display “Volume Down” when show thumb and one finger gesture	“Volume Down” is displayed after adjustment of gesture	Pass
		Display “Temperature Up” when show two finger gesture	“Temperature Up” is displayed	Pass
		Display “Temperature Down” when show one finger gesture	“Temperature Down” is displayed	Pass
9	Quit Program	Exit when “q” button is pressed	All window is terminated	Pass

Table 5.2.1-T1 Result of Black-Box Testing

Table 5.2.1-T1 shows the result of the black-box testing and all of the test cases have passed the test which indicate the system has met the functional requirements in the system design.

5.2.2 System Performance Testing

Table at below shows that the recognition result for each iteration and the average recognition rate (ARR) is being calculated in percentage from the recognition results for each gesture then ARR of the whole system will be computed.

i. System Performance in Room Environment

Gesture	**Recognition Result**										
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**ARR %**
 **Full Palm – Play**	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	100
 **Punch – Pause**	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	90

 **Thumb Left – Decline Call**	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	100
 **Thumb Right – Accept Call**	Y	N	Y	Y	Y	Y	Y	N	Y	Y	80
 **Thumb with Two Finger – Volume Up**	Y	Y	Y	Y	N	Y	Y	N	Y	Y	80

 **Thumb with One Finger – Volume Down**	N	Y	N	N	Y	Y	Y	Y	N	Y	60
 **Two Finger – Temperature Up**	Y	N	Y	Y	Y	Y	Y	N	Y	Y	80
 **One Finger – Temperature Down**	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	100

Table 5.2.2-T1 Result of System Performance Testing in Room Environment

Table 5.2.2-T1 shows the result of system performance testing in room environment. The recognition result is good as most of the gestures had achieved over 80% of ARR. Yet, there is one gesture that achieved only 60% of ARR and the result is consider undesirable for a real-time gesture recognition system that will be implemented in a car. Overall, the ARR for the whole system in recognizing all the gestures is at a satisfied level which is 86.25%.

ii. System Performance in Car Environment

Gesture	**Recognition Result**										
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**ARR %**
 **Full Palm – Play**	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	100
 **Punch – Pause**	Y	N	Y	Y	Y	Y	N	Y	Y	Y	80

 **Thumb Left – Decline Call**	N	Y	Y	Y	N	Y	Y	Y	N	Y	70
 **Thumb Right – Accept Call**	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	90
 **Thumb with Two Finger – Volume Up**	Y	N	Y	N	Y	Y	Y	N	Y	Y	70

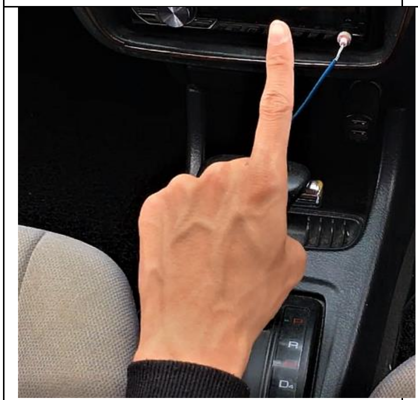
 **Thumb with One Finger – Volume Down**	N	Y	N	N	Y	Y	Y	N	Y	N	50
 **Two Finger – Temperature Up**	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	90
 **One Finger – Temperature Down**	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	90

Table 5.2.2-T2 Result of System Performance Testing in Car Environment

Table 5.2.2-T2 shows the result of system performance testing in car environment. The recognition result is still consider desirable as most of the gestures still have achieved more than 80% of ARR. Yet, there are few gestures are not getting a desirable ARR such as the gesture of decline call, volume up especially volume down are obtaining less than 70% of ARR. Overall, the ARR for the whole system in recognizing all the gestures is still at a satisfied level which is 80%. When compare to the system performance in room environment, the ARR of the system is slightly decrease as the room environment is using the optimized control factor and the car environment contain more uncertainties such as brightness of environment is not favourable and camera view may be not be at the best position.

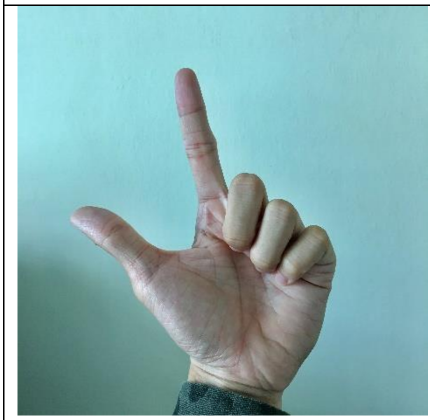
5.2.3 Classification Performance Testing

Table at below shows that the classification result for each iteration and the results are recorded in either True Positive (TP) or False Positive (FP). The result of classification for each gesture will be used to compute the accuracy and misclassification rate of the overall system.

i. Classification Performance in Room Environment

Gesture	Classification Result									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
 **Full Palm – Play**	TP	TP	TP	TP	TP	TP	TP	TP	TP	TP

 **Punch – Pause**	TP	TP	TP	TP	TP	TP	TP	TP	TP	TP
 **Thumb Left – Decline Call**	TP	TP	TP	TP	TP	TP	TP	TP	TP	TP
 **Thumb Right – Accept Call**	TP	TP	TP	TP	TP	TP	TP	TP	TP	TP

 **Thumb with Two Finger – Volume Up**	TP	FP	TP	TP	TP	TP	FP	TP	TP	TP
 **Thumb with One Finger – Volume Down**	FP	FP	TP	TP	TP	TP	FP	TP	FP	TP
 **Two Finger – Temperature Up**	TP	TP	TP	TP	TP	TP	TP	TP	TP	TP

 **One Finger – Temperature Down**	TP	TP	TP	TP	TP	TP	TP	TP	TP	TP

Table 5.2.3-T1 Result of Classification Performance Testing in Room Environment

Table 5.2.3-T1 shows the result of classification performance testing in room environment. Most of the gesture have achieved more than 80% of accuracy and the misclassification rate is relatively low. Yet, the classification result of the gesture “volume down” is not satisfied as it only achieved 60% of accuracy and a relative high misclassification rate of 40%. Overall, the classification performance of the real-time gesture recognition system in room environment is at a desirable level as the average accuracy is as high as 92.5% and the average misclassification rate is only 7.5%.

ii. Classification Performance in Car Environment

Gesture	Classification Result									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
 **Full Palm – Play**	TP	TP	TP	TP	TP	TP	TP	TP	TP	TP
 **Punch – Pause**	TP	TP	TP	TP	TP	TP	TP	TP	TP	TP
 **Thumb Left – Decline Call**	TP	TP	TP	TP	TP	TP	TP	TP	TP	TP

 **Thumb Right – Accept Call**	TP	TP	TP	TP	TP	TP	TP	TP	TP	TP
 **Thumb with Two Finger – Volume Up**	FP	TP	TP	FP	TP	TP	TP	TP	FP	TP
 **Thumb with One Finger – Volume Down**	TP	FP	FP	TP	TP	FP	TP	FP	FP	TP

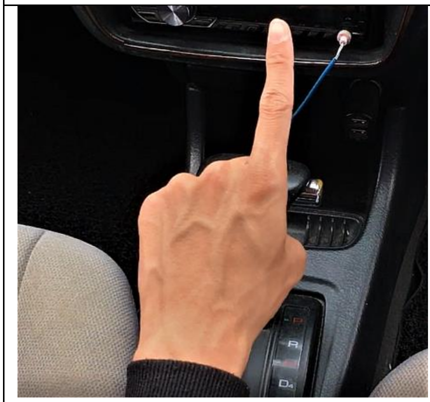
 **Two Finger – Temperature Up**	TP	TP	TP	TP	TP	TP	TP	TP	TP	TP
 **One Finger – Temperature Down**	TP	TP	TP	TP	TP	TP	TP	TP	TP	TP

Table 5.2.3-T2 Result of Classification Performance Testing in Car Environment

Table 5.2.3-T2 shows the result of classification performance testing in car environment. The classification result is desirable as most of the gesture have achieved 100% of accuracy. Only two gesture which is "Volume Up" and "Volume Down" have a lower accuracy of 70% and 50%. Overall, the classification performance of the real-time gesture recognition system in room environment is still at a satisfied level as the average accuracy is as high as 90% and the average misclassification rate is 10%.

By observing the classification performance in both room environment and car environment, two gestures are found lower in accuracy and higher in misclassification rate which is "Volume Up" and "Volume Down". This is because of the classification model which is the rules to determine gesture is not strong enough as the thumb is difficult to be detected when it is not fully extend and this problem will be recorded as part of the future work of the project.

CHAPTER 6: CONCLUSION

6.1 Conclusion

In a nutshell, the real-time gesture recognition system will be used to track and recognize several human static hand gestures by implementing several image processing techniques and algorithms that have been developed throughout the system development process. Furthermore, the system is able to simplify and enhance the interaction between human and computer because only natural mid-air hand gestures is used to interact with the system function which able to reduce driver distraction in the driving process. Unfortunately, the project is not able to achieve the initial project scope which is directly control the car infotainment function due to limited knowledge in the advanced automotive technology and limited resource in terms of cost and time. Therefore, the recognition result will be displayed only with the function name once the gesture is being recognized.

The system design is described using various diagrams which include block diagram, use-case diagram and activity diagrams that provide a clear picture of the overall system. Besides, Evolutionary Prototyping methodology is used to speed up the system development process and improve the quality of the final system. Moreover, the system is developed using high-level Python programming language which provides easy syntax that allow quick coding and it provides various standard libraries which enable the execution of complex functionalities easily. OpenCV open source library also being used for its various functions that related to object tracking and image processing.

The system process is separated into five stages where in the image acquisition stage, the user image is captured, resized to a fixed width, flipped the frame to avoid mirror view and set the ROI to minimize the recognition region. In the background subtraction stage, the foreground model is extracted and convert into HSV colour space then apply skin filter to extract skin region. Then the image will be transform into binary image using thresholding and lastly smooth out the image by morphological transformation. The next stage is the hand segmentation that perform contour detection and approximate the hand contour shape for features extraction. The features extraction is the core processing stage that extract the required features such as hand centre, fingertips, defect points, hull area, hand area and the angle of finger from sets of image processing algorithms and techniques. The last stage is the gesture recognition stage which use the extracted features to build the gesture recognition models that consist set of rules to recognize gestures.

Other than that, the system functionality, average recognition rate, accuracy and misclassification rate of the system is being evaluated in the system testing through functional testing and non-functional testing which include black-box testing, system performance testing and classification performance testing. From the black-box testing, all the test cases have passed which indicate the system has met the functional requirements and the project objectives.

For the system performance testing, the system is able to achieve a relatively satisfied of average recognition rate in both room and car environment. Yet, the system performance in the room environment is slightly higher than in the car environment due to the environment factors in the room environment is better than the car environment. For the classification performance testing, the classification result of most gestures is desirable in both room and car environment but just the classification result for the "Volume Up" and "Volume Down" gesture is lower in accuracy and contain high misclassification rate due to the weakness in the classification model. These weakness of the system will be recorded as part of the future work of the project in order to achieve better improvement.

6.2 Future Work

Currently the real-time gesture recognition system is still far away for a complete and perfect system because it didn’t able to perform its original intended function which is recognizing both static and dynamic hand gesture in directly control on the vehicle infotainment system function. Besides that, the system performance and classification performance of the system is just at the satisfied level and still required much improvement to maximize the average recognition rate and supress the error rate as much as possible in order to be implement in a real vehicle system with the standard of automotive industry.

In the future work, the system will consider using the better camera which able to collect the RGB and depth data from the captured image so that the system will not be restricted by the lightning condition and the clustered background issues in the gesture recognition process. In addition, the system can consider on implementing machine learning algorithms such as CNN, SVM and HMM in recognizing dynamic gesture as involved the temporal trajectory of some estimated parameter over time. Never the less, the classification models still have to be improved with the consideration of more effective rules in order to recognize gesture with minimal rate of error.

Eventually, the real-time gesture recognition system still requires a lot of improvement in order to meet the requirements and standards of ADAS.

REFERENCES

Akhtar, h., 2015. *Leap Motion – The Next Generation of Input Devices.* [Online] Available at: http://www.whitedust.net/leap-motion-the-next-generation-of-input-devices/ [Accessed 7 August 2018].

Amos, E., 2011. *Kinect for Xbox 360. The Xbox 360 E revision has an Xbox logo to the left of the Xbox 360 branding..* [Online] Available at: https://en.wikipedia.org/wiki/Kinect#/media/File:Xbox-360-Kinect-Standalone.png [Accessed 7 August 2018].

Bear, j. H., 2018. *What Is the HSV Color Model?.* [Online] Available at: https://www.lifewire.com/what-is-hsv-in-design-1078068 [Accessed 7 August 2018].

Blundell, B. J., 2011. *Skin Segmentation.* [Online] Available at: https://www.section9.co.uk/posts/2011-07-28-Skin.html [Accessed 7 August 2018].

Business Wire, 2016. *Top 3 Trends Impacting the Global Automotive Advanced Driver Assistance System Market Through 2020: Technavio.* [Online] Available at: https://www.businesswire.com/news/home/20161102006225/en/Top-3-Trends-%20Impacting-Global-Automotive-Advanced [Accessed 2017 November 2017].

Chen, Q., 2008. Real-Time Vision-Based Hand Tracking and Gesture Recognition. *Doctoral Dissertation.*

Cuyugan, M., 2018. *ASUS TUF FX504 gaming laptop.* [Online] Available at: https://www.techpatrl.com/asus-tuf-fx504-gaming-laptops-official/ [Accessed 7 8 2018].

Data School, 2014. *Simple guide to confusion matrix terminology.* [Online] Available at: https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/
[Accessed 13 August 2018].

Definition, N., 2013. *Opencv python hand gesture recognition.* [Online] Available at: http://creat-tabu.blogspot.com/2013/08/opencv-python-hand-gesture-recognition.html
[Accessed 7 August 2018].

Dondi, P., Lombardi, L. & Porta, M., 2014. Development of Gesture-based Human Computer Interaction Application by Fusion of Depth and Colour Video Stream. *IET Computer Vision,* 8(6), pp. 568-578.

Doxygen, 2015. *Contour Features.* [Online]
Available at:
https://docs.opencv.org/3.1.0/dd/d49/tutorial_py_contour_features.html
[Accessed 7 August 2017].

Doxygen, 2017. *Image thresholding.* [Online]
Available at:
https://docs.opencv.org/3.4.0/d7/d4d/tutorial_py_thresholding.html
[Accessed 7 August 2018].

Doxygen, 2018. *How to Use Background Subtraction Methods.* [Online]
Available at:
https://docs.opencv.org/3.4.1/d1/dc5/tutorial_background_subtraction.html
[Accessed 7 August 2018].

Euda, E., Matsumoto, Y., Imai, M. & Ogasawara, T., 2003. A Hand Pose Estimation for Vision-Based Human Interfaces. *IEEE Trans. Ind. Electron,* 50(4), pp. 676-684.

Hofmann, F. G., Heyer, P. & Hommel, G., 1998. Velocity profile based recognition of dynamic gestures with discrete Hidden Markov Models. *Gesture and Sign Language in Human-Computer Interaction,* Volume 1371, pp. 81-95.

Jetbrains, 2016. *PyCharm Logo.* [Online]
Available at:
https://en.wikipedia.org/wiki/PyCharm#/media/File:PyCharm_Logo.svg
[Accessed 28 February 2018].

JetBrains, n.d. *PyCharm.* [Online]
Available at: https://www.jetbrains.com/pycharm/features/
[Accessed 23 February 2018].

Jolliffe, I. T., 2002. Principal Component Analysis. In: New York: Springer Science & Business Media.

Marilly, E., Gonguet, A., Martinot, O. & Pain, F., 2013. Gesture Interaction with Video: From Algorithms to User Evaluation. *Bell Labs Technical Journal*, 17(4), pp. 103-118.

OpenCV dev team, 2014. *Background Subtraction.* [Online]
Available at: https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_video/py_bg_subtraction/py_bg_subtraction.html
[Accessed 7 August 2018].

OpenCV dev team, 2014. *Changing Colorspaces.* [Online]
Available at: https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_colorspaces/py_colorspaces.html
[Accessed 7 August 2018].

OpenCV dev team, 2014. *Morphological Transformations.* [Online]
Available at: https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html
[Accessed 7 August 2018].

Oregonstate.edu, 2017. *"Distracted driving" at an all-time high; new approaches needed | News and Research Communication..* [Online]
Available at:
http://oregonstate.edu/ua/ncs/archives/2015/mar/%E2%80%9Cdistracted-driving%E2%80%9D-all-time-high-new-approaches-needed
[Accessed 24 November 2017].

Partners, W. C., 2016. *Beyond The Headlights: ADAS and Autonomous Sensing,* Palo Alto, CA: s.n.

Plouffe, G. & Cretu, A. M., 2016. Static and Dynamic Hand Gesture Recognition in Depth Data Using Dynamic Time Wraping. *IEEE Transactions on Instrumentation and Measurement,* 65(2), pp. 305-316.

Popov, P., 2013. *GestureVue: A human computer interaction project.* [Online] Available at: http://www.site.uottawa.ca/~laganier/projects/facegesture/ [Accessed 7 August 2018].

Rossol, N., Cheng, I. & Basu, A., 2016. A Multisensor Technique for Gesture Recognition Through Intelligent Skeletal Pose Analysis. *IEEE Transactions on Human-Machine System,* 46(3), pp. 350-359.

Shaik, K. B., Ganesan, P., Kalist, V. & Sathish, B. S., 2015. Comparative Study of Skin Color Detection and Segmentation in HSV and YCbCr Color Space. *Procedia Computer Science,* Volume 57, pp. 41-48.

Shapiro, L. G. & Stockman, G. C., 2001. Computer Vision. *Computer Vision,* pp. 279-325.

Shavit, A., 2006. *OpenCV Logo.* [Online] Available at: http://opencv.willowgarage.com/wiki/OpenCVLogo [Accessed 28 February 2018].

SmartDraw, n.d. *UML Diagram.* [Online] Available at: https://www.smartdraw.com/uml-diagram/#whatIsUML [Accessed 25 February 2018].

Sommerville, I., 2000. *Software engineering.* 6th ed. s.l.:Addison-Wesley.

TutorialPoint, n.d. *UML - Standard Diagrams.* [Online] Available at: https://www.tutorialspoint.com/uml/uml_standard_diagrams.htm [Accessed 25 February 2018].

Wachs, J. P., Kölsch, M., Stern, H. & Edan , Y., 2011. Vision-based hand-gesture applications. *Communications of the ACM,* 54(2), pp. 60-71.

Waterdown Collision, 2017. *The Leading Causes of Car Accidents - Causes and Statistics.*. [Online]
Available at: http://www.waterdowncollision.com/blog/safe-driving/leading-causes-of-car-accidents/
[Accessed 24 November 2017].

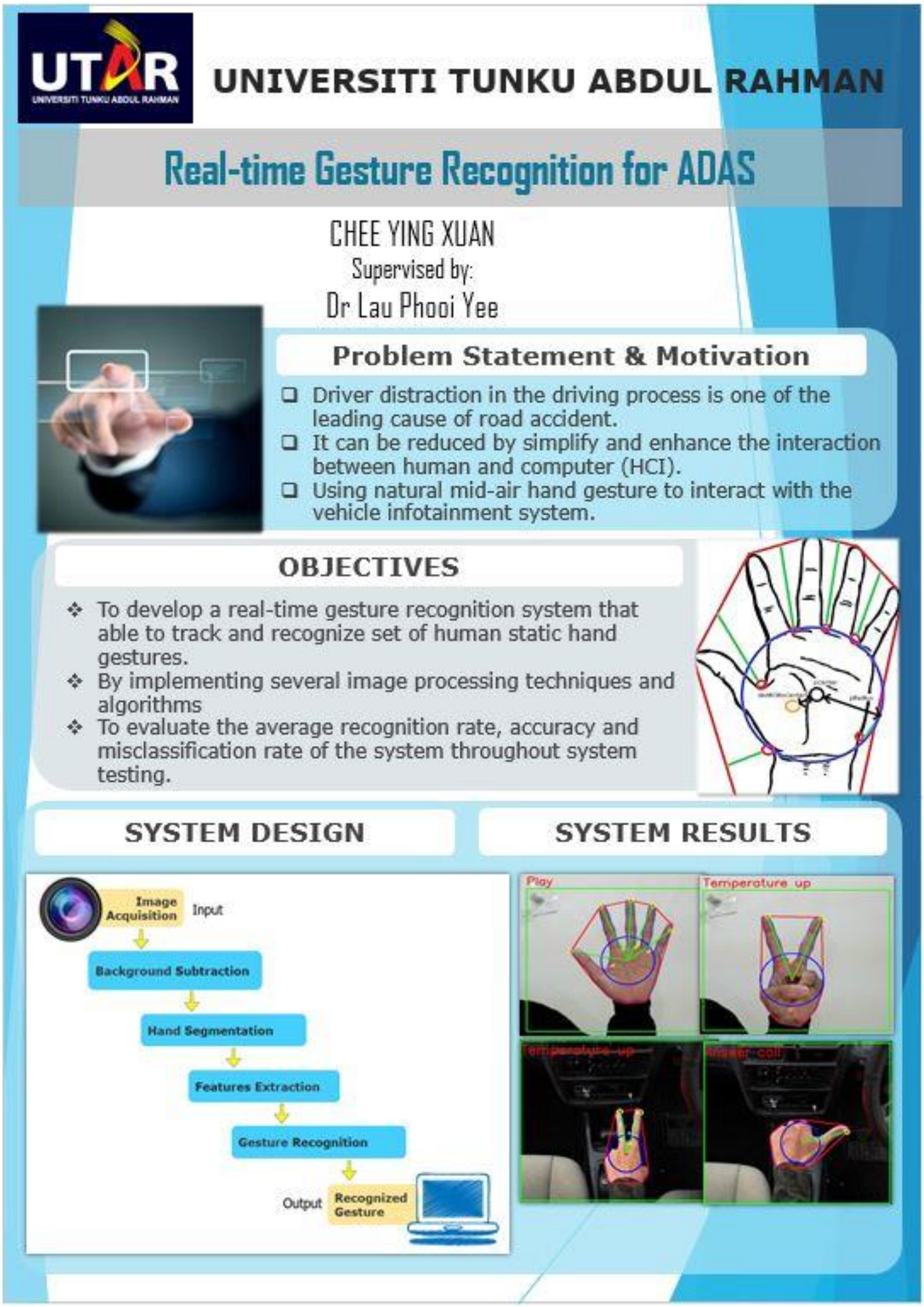
Weinberg, J., n.d. *Evolutionary Prototyping Model.* [Online]
Available at: https://hci.cs.siue.edu/NSF/Files/Semester/Week13-1/PPT-Text/Slide12.html
[Accessed 25 February 2018].

Wu, G. & Kang, W., 2016. Robust Fingertip Detection in a Complex Environment. *IEEE TRANSACTIONS ON MULTIMEDIA,* Volume 18, pp. 978-987.

www.python.org, 2008. *Python logo and wordmark.* [Online]
Available at:
https://en.wikipedia.org/wiki/File:Python_logo_and_wordmark.svg
[Accessed 28 February 2018].

Yao, Y. & Fu, Y., 2014. Contour Model-Based Hand Gesture Recognition Using the Kinect Sensor. *IEEE Transactions on Circuits and Systems for Video technology,* 24(11), pp. 1935-1944.

POSTER


UNIVERSITI TUNKU ABDUL RAHMAN
Real-time Gesture Recognition for ADAS
CHEE YING XUAN
Supervised by:
Dr Lau Phooi Yee
Problem Statement & Motivation
Driver distraction in the driving process is one of the leading cause of road accident.
It can be reduced by simplify and enhance the interaction between human and computer (HCI).
Using natural mid-air hand gesture to interact with the vehicle infotainment system.
OBJECTIVES
To develop a real-time gesture recognition system that able to track and recognize set of human static hand gestures.
By implementing several image processing techniques and algorithms
To evaluate the average recognition rate, accuracy and misclassification rate of the system throughout system testing.
SYSTEM DESIGN
Image Acquisition
Input
Background Subtraction
Hand Segmentation
Features Extraction
Gesture Recognition
Output
Recognized Gesture
SYSTEM RESULTS
Play
Temperature up

APPENDICES

FINAL YEAR PROJECT WEEKLY REPORT

*(Project I / **Project II**)*

Trimester, Year: May, 2018	**Study week no.: 3**
Student Name & ID: Chee Ying Xuan 140ACB03243	
Supervisor: Dr. Lau Phooi Yee	
Project Title: Real-time Gesture Recognition System for ADAS	

1. WORK DONE Previous work from Project 1
2. WORK TO BE DONE Image acquisition and background subtraction
3. PROBLEMS ENCOUNTERED Problem on background subtraction, can't track hand properly in clustered background and unstable lightning
4. SELF EVALUATION OF THE PROGRESS Slow progress, require catch up quickly

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

*(Project I / **Project II**)*

Trimester, Year: May, 2018	**Study week no.: 5**
Student Name & ID: Chee Ying Xuan 140ACB03243	
Supervisor: Dr. Lau Phooi Yee	
Project Title: Real-time Gesture Recognition System for ADAS	

1. WORK DONE Background subtraction problem solved
2. WORK TO BE DONE Hand segmentation and features subtraction, looking forward for a way to locate fingertips
3. PROBLEMS ENCOUNTERED Unable to display hand features properly on its original coordinates
4. SELF EVALUATION OF THE PROGRESS Slow progress on the system due to search for the proper fingertips detection algorithm

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

*(Project I / **Project II**)*

Trimester, Year: May, 2018	Study week no.: 6
Student Name & ID: Chee Ying Xuan 140ACB03243	
Supervisor: Dr. Lau Phooi Yee	
Project Title: Real-time Gesture Recognition System for ADAS	

1. WORK DONE Image acquisition, background subtraction, hand segmentation
2. WORK TO BE DONE Completing the features extraction part
3. PROBLEMS ENCOUNTERED The fingertips position of thumb is too near to the centre and it is hard to be detected
4. SELF EVALUATION OF THE PROGRESS Moderate progress

_________________________ _________________________

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

*(Project I / **Project II**)*

Trimester, Year: May, 2018	**Study week no.: 8**
Student Name & ID: Chee Ying Xuan 140ACB03243	
Supervisor: Dr. Lau Phooi Yee	
Project Title: Real-time Gesture Recognition System for ADAS	

1. WORK DONE Until features extraction is completed, yet still requires enhancement
2. WORK TO BE DONE Design gesture recognition model, set of rules to recognize gesture using features extracted
3. PROBLEMS ENCOUNTERED Some gestures having similar features and hard to be differentiated, requires change on some gesture
4. SELF EVALUATION OF THE PROGRESS Moderate progress due to busy schedule

____
Supervisor's signature

____
Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

*(Project I / **Project II**)*

Trimester, Year: May, 2018	**Study week no.: 9**
Student Name & ID: Chee Ying Xuan 140ACB03243	
Supervisor: Dr. Lau Phooi Yee	
Project Title: Real-time Gesture Recognition System for ADAS	

1. WORK DONE Part of the gesture recognition model is done, just left with some gesture that requires modification
2. WORK TO BE DONE Complete gesture recognition model for all the gesture and optimize the code to achieve better performance
3. PROBLEMS ENCOUNTERED The gesture punch is hard to be detected as the system will wrongly detect some fingertips
4. SELF EVALUATION OF THE PROGRESS Moderate progress due to busy schedule

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

*(Project I / **Project II**)*

Trimester, Year: May, 2018	**Study week no.: 11**
Student Name & ID: Chee Ying Xuan 140ACB03243	
Supervisor: Dr. Lau Phooi Yee	
Project Title: Real-time Gesture Recognition System for ADAS	

1. WORK DONE Whole gesture recognition system is done
2. WORK TO BE DONE Design the testing plan and working on the documentation
3. PROBLEMS ENCOUNTERED Some of the predetermined evaluation plan is unable to be implemented, requires modification
4. SELF EVALUATION OF THE PROGRESS Moderate progress, rushing on the report

_________________________ _________________________

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

*(Project I / **Project II**)*

Trimester, Year: May, 2018	**Study week no.: 12**
Student Name & ID: Chee Ying Xuan 140ACB03243	
Supervisor: Dr. Lau Phooi Yee	
Project Title: Real-time Gesture Recognition System for ADAS	

1. WORK DONE The whole system and the system testing is done
2. WORK TO BE DONE Finalize on the project report
3. PROBLEMS ENCOUNTERED Formatting issues
4. SELF EVALUATION OF THE PROGRESS Moderate progress but still on schedule

Supervisor's signature

Student's signature

Real-time Gesture Recognition for ADAS

ORIGINALITY REPORT

12%	6%	9%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.utar.edu.my Internet Source	1%
2	"Computer Vision and Machine Learning with RGB-D Sensors", Springer Nature America, Inc, 2014 Publication	1%
3	Wu, Guile, and Wenxiong Kang. "Robust Fingertip Detection in a Complex Environment", IEEE Transactions on Multimedia, 2016. Publication	1%
4	Rossol, Nathaniel, Irene Cheng, and Anup Basu. "A Multisensor Technique for Gesture Recognition Through Intelligent Skeletal Pose Analysis", IEEE Transactions on Human-Machine Systems, 2015. Publication	1%
5	"Gesture Interactions With Video: From Algorithms to User Evaluation", Bell Labs Technical Journal, 2013. Publication	1%

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

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Chee Ying Xuan
ID Number(s)	14ACB03243
Programme / Course	Information Systems Engineering
Title of Final Year Project	Real-time Gesture Recognition System for ADAS

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: __12__ **%** **Similarity by source** Internet Sources: __6__ % Publications: __9__ % Student Papers: __5__ %	
Number of individual sources listed of more than 3% similarity: __0__	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

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

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

______________________________ ______________________________

Signature of Supervisor | Signature of Co-Supervisor

Name:__________________________ | Name:__________________________

Date: ___________________________ | Date: ___________________________

UNIVERSITI TUNKU ABDUL RAHMAN
FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	14ACB03243
Student Name	Chee Ying Xuan
Supervisor Name	Dr. Lau Phooi Yee

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Cover
	Signed Report Status Declaration Form
	Title Page
	Signed form of the Declaration of Originality
	Acknowledgement
	Abstract
	Table of Contents
	List of Figures (if applicable)
	List of Tables (if applicable)
	List of Symbols (if applicable)
	List of Abbreviations (if applicable)
	Chapters / Content
	Bibliography (or References)
	All references in bibliography are cited in the thesis, especially in the chapter of literature review
	Appendices (if applicable)
	Poster
	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

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

I, the author, have checked and confirmed all the items listed in the table are included in my report. ______________________ (Signature of Student) Date:	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction. ______________________ (Signature of Supervisor) Date: