HAND GESTURE RECOGNITION SYSTEM

Submitted By

MUHAMMAD ARSLAN

CS181101

SHAHERYAR AKBAR

CS181102

(2018-2022)

Department of Computer Science

Khawaja Fareed University of Engineering & Information Technology

Rahim Yar Khan

2022

HAND GESTURE RECOGNITION SYSTEM

Submitted to

Mr. Shahzad Hussain

Department of Computer Science

In partial fulfilment of the requirements

For the degree of

Bachelor of Science in Computer Science

By

MUHAMMAD ARSLAN

CS181101

SHAHERYAR AKBAR

CS181102

2018-2022

Khawaja Fareed University of Engineering & Information Technology

Rahim Yar Khan

2022

DECLARATION

We hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. We also declared that it has not been previously and concurrently submitted for any other degree or award at Khawaja Fareed University of engineering & Information Technology or other institutions.

Reg No : ________________________	Reg No : ________________________
Name : ________________________	Name : ________________________
Signature : ________________________	Signature : ________________________
Date : ________________________	Date : ________________________

APPROVAL FOR SUBMISSION

I certify that this project report entitled "**HAND GESTURE RECOGNITION SYSTEM**" was prepared by **MUHAMMAD ARSLAN & SHAHERYAR AKBAR** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science in Computer Science at Khawaja Fareed University of Engineering & Information Technology.

Approved by:

Signature : ___________________________

Supervisor : Mr. Shahzad Hussain

Date : ___________________________

MEETING LOG

Project Title	**Hand Gesture Recognition System**
Name of Student(s) & Registration#	Muhammad Arslan (CS181101) Shaheryar Akbar (CS181102)
Name of Supervisor	Mr. Shahzad Hussain

Meeting#	Date --/--/----	Topic of Discussion/Task	Supervisor's Signature
01	10-22-2021	Discuss about the module of image processing / Read the instruction about the image processing.	
02	10-29-2021	Discuss about the module of hand tracking / Read the instruction about the Mediapipe module.	
03	11-05-2021	Discusses about the color conversion / Converting color conversion BGR to RGB.	
04	11-12-2021	Discusses about the mouse movement / Find out the module for mouse movement	
05	11-26-2021	Discuss about the keyboard and mouse functionality / find article and module keyboard and mouse functionality	
06	12-03-2021	Discuss about the width and height of new window when code in running form / Read article about the height and width size.	
07	12-10-2021	Discuss about the gesture / Find out the total gesture according to our project.	
08	12-17-2021	Discuss about the problem in pycharm / Read about the python Interpreter	
09	12-24-2021	Checking some functionality of running code / train data in better performance	
10	12-31-2021	Discusses about the play and pause function / Play and pause video with the help of hand gesture.	
11	01-7-2022	Discusses about the overall project / Finalize the project and documentation	
12	01-14-2022	Project is completed discusses about the performance and documentation.	

ACKNOWLEDGEMENT

We would like to thank everyone who had contributed to this project. We would like to express our gratitude to our Project supervisor, Mr. Shahzad Hussain for his invaluable advice, guidance, and his enormous patience throughout the development of the project.

In addition, we would also like to express our gratitude to our loving parents and friends who had helped and given us encouragement.

ABSTRACT

Recently, vision-based hand gesture recognition (HGR) has emerged as a natural and flexible human-computer interaction (HCI) approach. Users can control smart devices by applying hand gestures to imagers. However, prior efforts suffer from various limitations, such as excessive power consumption, low accuracy, and poor flexibility. The 3D HGR processors - [2] suffer from extremely large power overhead due to the employment of complex image processing, for example using Convolutional Neural Networks (CNNs). The grayscale sensor-based SoC [3] consumes less power. However, its accuracy is compromised, especially when the contrast between a hand gesture and the background is low. The infrared sensor-based SoC [4] can recognize 8 dynamic gestures with high accuracy (96%). Nevertheless, the over-simplified algorithm requires hand motion with a fixed gesture type, which limits the number of recognized dynamic gestures. Therefore, an ultra-low-power, flexible, and high accuracy HGR system is required.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Hands are human organs which are used to manipulate physical objects. For this very reason hands are used most frequently by human beings to communicate and interact with machines.

- Mouse and Keyboard are the basic input/output to computers and the use of both of these devices require the use of hands.
- Most important and immediate information exchange between man and machine is through visual and aural aid, but this communication is one sided.
- Computers of this age provide humans with 1024 * 768 pixels at a rate of 15 frames per second and compared to it a good typist can write 60 words per minute with each word on average containing 6 letters.
- To help somewhat mouse remedies this problem, but there are limitations in this as well.
- Although hands are most commonly used for day to day physical manipulation related tasks, but in some cases they are also used for communication.
- Hand gestures support us in our daily communications to convey our messages clearly.
- Hands are most important for mute and deaf people, who depends their hands and gestures to communicate, so hand gestures are vital for communication in sign language.
- If computer had the ability to translate and understand hand gestures, it would be a leap forward in the field of human computer interaction.
- The dilemma, faced with this is that the images these days are information rich and in-order to achieve this task extensive processing is required.
- Every gesture has some distinct features, which differentiates it from other gestures.
- Real life applications of gesture based human computer interaction are; interacting with virtual objects, in controlling robots, translation of body and sign language and controlling machines using gestures.

1.2 Image Processing

Image processing is reckoned as one of the most rapidly involving fields of the software industry with growing applications in all areas of work. It holds the possibility of developing the ultimate machines in future, which would be able to perform the visual function of living beings. As such, it forms the basis of all kinds of visual automation.

1.3 Hand Gesture Detection and Recognition

1.3.1 Detection

Hand detection is related to the location of the presence of a hand in a still image or sequence of images i.e. moving images. In case of moving sequences it can be followed by tracking of the hand in the scene but this is more relevant to the applications such as sign language. The underlying concept of hand detection is that human eyes can detect objects which machines cannot with that much accuracy as that of a human. From a machine point of view, it is just like a man fumble around with his senses to find an object.

The factors, which make the hand detection task difficult to solve, are:

1.3.1.1 Variations in image plane and pose

The hands in the image vary due to rotation, translation and scaling of the camera pose or the hand itself. The rotation can be both in and out of the plane.

1.3.1.2 Skin Color and Other Structure Components

The appearance of a hand is largely affected by skin color, size and also the presence or absence of additional features like hairs on the hand further adds to this variability.

1.3.1.3 Lighting Condition and Background

As shown in Figure 1.1 light source properties affect the appearance of the hand. Also the background, which defines the profile of the hand, is important and cannot be ignored.

Figure 1.1 Lighting Condition and Background

1.3.2 Recognition

Hand detection and recognition have been significant subjects in the field of computer vision and image processing during the past 30 years. There have been considerable achievements in these fields and numerous approaches have been proposed.

1.4 Problem Statement

“Hand Gesture Recognition Using Camera” is based on concept of Image processing. In recent year there is lot of research on gesture recognition using Kinect sensor on using HD camera but camera and Kinect sensors are costlier. This paper is focus on reduce cost and improve robustness of the proposed system using simple web camera.

1.5 Objectives

To build a system which recognizes the real time gesture display by user using Histogram-approach model. Build a such system which can perform multiple functionality of systems by gesturing. without using keyboard and mouse and perform their functionality very well.

- Selection of files
- Drag and drop files
- Control multimedia
- Sign Language

1.6 Project Scope

The scope of this project is to build a real time gesture classification system that can automatically detect gestures in natural lighting condition. In order to accomplish this objectives, a real time gesture based system is developed to identify gestures.
This system will work as one of futuristic of Artificial Intelligence and Computer Vision with user interface. Its create method to recognize hand gesture based on different parameters. The main priority of this system is to simple, easy and user friendly without making any special hardware. All computation will occur in single PC or workstation. Only special hardware will use to digitize the image (Webcam).

1.7 Advantages of Proposed Solution

- Gesture are easier representation, makes the presentation attractive, Quick expressing of message, etc.
- Gestures are non-verbal communications.
- It can make the information to be presented easily via audio, visual, or even through silent.
- It is usually a substitute of verbal based communication.
- People can easily interpret the gesture of another person.
- Gestures are the main mode of communication hearing impaired persons.

1.8 Relevance to Study Program

This project is related to my study program. In which program BSCS we studied different languages and tools related to hand detection and hand recognition techniques.

1.8.1 Python

Python is a common and simple language. A good read Python program sounds like reading English (but very strict English!). This pseudo-code nature of Python is its greatest strength. It allows you to concentrate on the solution to the problem rather than the syntax i.e. the language itself.

1.8.2 OpenCV

Open Source Computer Vision Library is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including features like hand detection and object detection.

1.8.3 Deep-Learning/AI

Artificial intelligence is the simulation of human intelligence through machines, especially computer systems. These processes include learning (rules for obtaining and using information), reasoning (using rules to make an estimate or final conclusion), and self-correction. AI's special applications include expert systems, hand recognition and machine vision.

1.9 Chapter Summary

After introducing the background of the problem, the argument of this project came up. The chapter describes the objectives, advantages, scope and use of the project. In addition, the chapter describes the team members and their contributions to the development of the project, as presented in a later report.

CHAPTER 2

LITERATURE REVIEW

2.1 Literature Review

Hand gesture recognition research is classified in three categories.

- First **“Glove based Analysis”** attaching sensor with gloves mechanical or optical to transduces flexion of fingers into electrical signals for hand posture determination and additional sensor for position of the hand. This sensor is usually an acoustic or a magnetic that attached to the glove. Look-up table software toolkit provided for some applications to recognize hand posture [3].
- The second approach is **“Vision based Analysis”** that human beings get information from their surroundings, and this is probably most difficult approach to employ in satisfactory way. Many different implementations have been tested so far. One is to deploy 3-D model for the human hand. Several cameras attached to this model to determine parameters corresponding for matching images of the hand, palm orientation and joint angles to perform hand gesture classification. Lee and Kunii developed a hand gesture analysis system based on a three-dimensional hand skeleton model with 27 degrees of freedom. They incorporated five major constraints based on the human hand kinematics to reduce the model parameter space search. To simplify the model matching, specially marked gloves were used [3].
- The Third implementation is **“Analysis of drawing gesture”** use stylus as an input device. These drawing analysis lead to recognition of written text. Mechanical sensing work has used for hand gesture recognition at vast level for direct and virtual environment manipulation. Mechanically sensing hand posture has many problems like electromagnetic noise, reliability and accuracy. By visual sensing gesture interaction can be made potentially practical but it is most difficult problem for machines [1, 2].

Excellent work has been done in support of machine sign language recognition by Sperling and Parish, who has done careful studies on the bandwidth necessary for a sign conversation using spatially and temporally sub-sampled images. Point light experiments (where “lights” are attached to significant locations on the body and just these points are used for recognition), have been carried out by Poizner. Most systems to date study isolate/static gestures. In most of the cases those are fingerspelling signs.

2.2 Methodology

Hand gesture recognition project is done using Python programming language and OpenCV as library. Python programming language produces simple and easy system code to understand. Also, Python package used here is Numpy. The image that is captured using web camera will be processed in a region called as Region of Interest (ROI) where act as a region of wanted area while ignoring the outside region, called background.

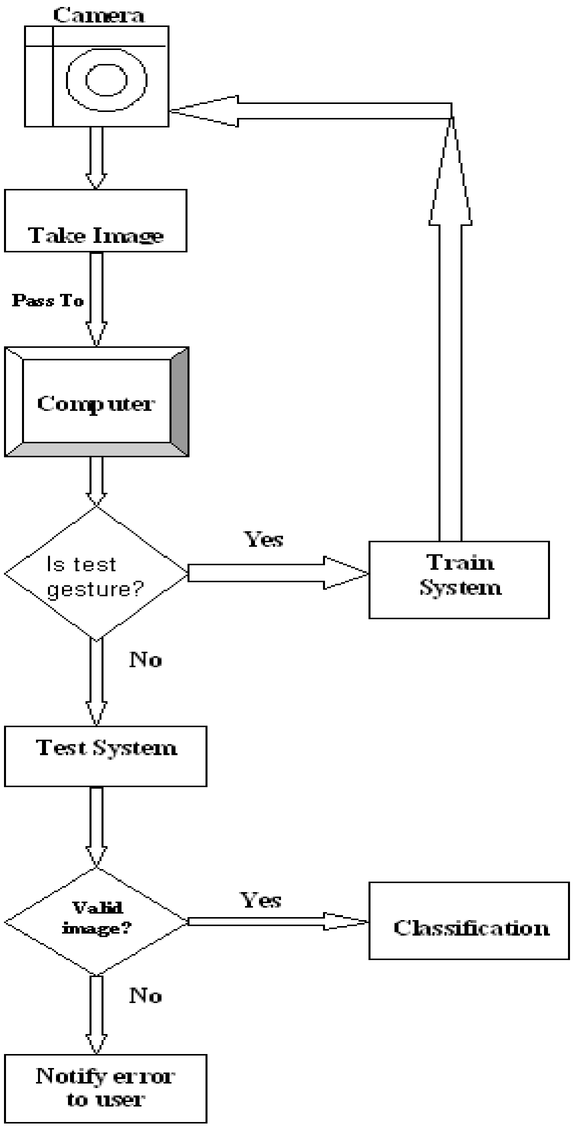

Figure 2.1 Methodology Diagram

2.3 Camera Orientations and Distance

It is very important to careful about direction of camera to permit easy choice of background. Two good and more effective approaches are to point the camera towards wall or floor. Lighting was standard room; intensity of light would be higher and shadowing effects lower because camera was pointed downwards. The distance of the camera from the hand should be such that it covers the entire gesture mainly.

2.4 Background Selection

Another important aspect is to maximize differentiation that the color of background must be different as possible from skin color. The floor color in the work used was black. It was decided to use this color because it offered minimum self-shadowing problem as compared to other background colors.

2.5 Different Recognition Approaches

The different recognition approaches studied are as follows:

2.5.1 Pen-Based Gesture Recognition

Recognizing gestures from two-dimensional input devices such as a pen or mouse has been considered for some time. The early Sketchpad system in 1963 used light-pen gestures, for example. Some commercial systems have used pen gestures since the 1970s. There are examples of gesture recognition [3] for document editing for air traffic control, and for design tasks such as editing splines.

There have been commercially available Personal Digital Assistants (PDAs) for several years, starting with the Apple Newton, and more recently the 3Com Palm Pilot and various Windows CE devices, Long, and Rowe survey problems and benefits of these gestural interfaces and provide insight for interface designers.

2.5.2 Tracker-Based Gesture Recognition

There are many tracking systems available commercially which can used for gesture recognition, primarily tracking eye gaze, hand gesture [3], and overall body and its position. In virtual environment interaction each sensor has its own strengths and weaknesses. Gestural interface eye gaze can be useful, so I focus here on gesture based input from tracking the hand and the body.

2.5.3 Body Suits

Process of small place of strategically dots placed on human body, people can perceive patterns such as gestures, activities, identities and other aspects of body. One way of approach is recognition of postures and human movements is optically measure of 3D position such as markers attached to body and then recovers time varying articulate structure of body. This articulated sensing by position and joint angles using electromechanically sensors. Although some of system require small ball or dot placed top user clothing. I prefer body motion capture by “body suits” generically.

Body suits have **advantages and disadvantages** similar to those of data gloves. At high sampling rate it provides reliable results but they are cumbersome and very expensive. Non-trivial calibration. Several cameras used by optical system which is typically offline data process, lack of wires and tether is major disadvantages [2].

2.5.4 Hand and Arm Gestures

These two parts of body (Hand & Arm) have most attention among those people who study gestures in fact much reference only consider these two for gesture recognition. The majority of automatic recognition systems are for deictic gestures (pointing), emblematic gestures (isolated signs) and sign languages (with a limited vocabulary and syntax). Some are components of bimodal systems, integrated with speech recognition. Some produce precise hand and arm configuration while others only coarse motion [3].

2.5.5 Vision-Based Gesture Recognition

The most **significant disadvantage** of the tracker-based systems is that they are **cumbersome**. This detracts from the immerse nature of a virtual environment by requiring the user to put on an unnatural device that cannot easily be ignored, and which often requires significant effort to put on and calibrate. Even optical systems with markers applied to the body suffer from these shortcomings, albeit not as severely. What many have wished for is a technology that provides real-time data useful for analyzing and recognizing human motion that is passive and non-obtrusive. Computer vision techniques have the potential to meet these requirements.

This technique was also used by us for recognizing hand gestures in real time. With the help of a web camera, I took pictures of hand on a prescribed background and then applied the classification algorithm for recognition [3].

2.6 Chapter Summary

In this chapter we discussed about the literature review based on the previous research. In which we discussed about the problems of hand gesture recognition system and methodology of hand gesture recognition system and also discusses about the different recognition approaches.

CHAPTER 3

REQUIREMENT ENGINEERING

3.1 Hand Detection

The original images used for hand gesture recognition in the work are demonstrated. These images are captured with a normal camera. These hand images are taken under the same condition. The background of these images is identical. So, it is easy and effective to detect the hand region from the original image using the background subtraction method. However, in some cases, there are other moving objects included in the result of background subtraction. The skin color can be used to discriminate the hand region from the other moving objects. The color of the skin is measured with the HSV model. The HSV (hue, saturation, and value) value of the skin color is 315, 94, and 37, respectively. The image of the detected hand is resized to 200×200 to make the gesture recognition invariant to image scale.

Figure 3.1 The procedure of hand detection

3.2 Fingers and Palm Segmentation

The output of the hand detection is a binary image in which the white pixels are the members of the hand region, while the black pixels belong to the background. An example of

Figure 3.2 The detected hand region.

the hand detection result is shown in **Figure 1.8.** Then, the following procedure is implemented on the binary hand image to segment the fingers and palm.

3.3 Fingers Recognition

In the segmentation image of fingers, the labeling algorithm is applied to mark the regions of the fingers. In the result of the labeling method, the detected regions in which the number of pixels is too small is regarded as noisy regions and discarded. Only the regions of enough sizes are regarded as fingers and remain. For each remained region, that is, a finger, the minimal bounding box is found to enclose the finger.

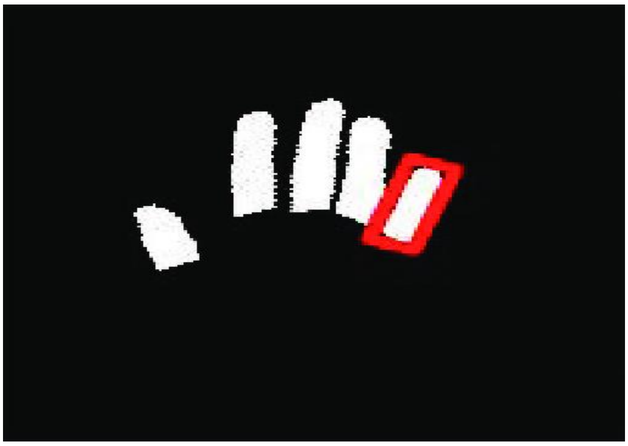

Figure 3.3 The minimal bounding box.

3.4 Recognition of Hand Gestures

When the fingers are detected and recognized, the hand gesture can be recognized using a simple rule classifier. In the rule classifier, the hand gesture is predicted according to the number and content of fingers detected. The content of the fingers means what fingers are detected. The rule classifier is very effective and efficient.

Figure 3.4 The image set of hand gestures used in the experiments.

3.5 Understanding the System

Hand gesture recognition system has evolved tremendously in the recent few years because of its ability to interact with machine efficiently. Mankind tries to incorporate human gestures into modern technology by searching and finding a replacement of multi touch technology which does not require any touching movement on screen.

3.5.1 User Involvement

The involvement of the user in this system is main because our system is operating by using hand gesture of the user.

3.5.2 Stakeholders

Stakeholders are main user of project. They can manage their system in a good manner, they can easily use the system by the help of hand gesture.

3.5.3 Domain

we are utilizing our final year project in different areas and different types of people. Our project will be used in different gaming industry also it can be used for dumb person with the help of sign language.

3.6 Requirement Engineering

3.6.1 Functional Requirement

Functional requirements specify the main technical functionalities and specifications that the system should incorporate.

Table 3.1 Functional Requirement 01

Functional Requirement ID	**FR-01**
Functions	**Webcam Start**
Description	System webcam open
Input Source	User Run the code
Output Source	Histogram will be shown
Pre-Condition	Webcam histogram display
Post Condition	Webcam open

Table 3.2 Functional Requirement 02

Functional Requirement ID	**FR-02**
Functions	**Capture Video**
Description	Webcam capture video
Input Source	System will capture video
Output Source	Video will be captured
Pre-Condition	User will show hand in front of webcam

Post Condition	Webcam capture the video of hand

Table 3.3 Functional Requirement 03

Functional Requirement ID	**FR-03**
Functions	**Capture Gesture**
Description	Gesture will show by User
Input Source	User show hand in front of Webcam
Output Source	Gesture will be capture
Pre-Condition	User will show gesture with the help of hand
Post Condition	System take gesture and process it.

Table 3.4 Functional Requirement 04

Functional Requirement ID	**FR-04**
Functions	**Translate Gesture**
Description	Gesture translated
Input Source	User gesture can be translating
Output Source	Gesture can be translated
Pre-Condition	User express the gesture to the system
Post Condition	System will go the class of gesture will be express by user

Table 3.5 Functional Requirement 05

Functional Requirement ID	**FR-05**
Functions	**Extract Features**
Description	Extract the feature of hands
Input Source	Feature of hands
Output Source	Feature can be extracted
Pre-Condition	Reduce the number of features in dataset
Post Condition	Creating new features from existing ones.

Table 3.6 Functional Requirement 06

Functional Requirement ID	**FR-06**
Functions	**Match Features**
Description	Features can be matched the given features
Input Source	Giving feature for matching
Output Source	Features can be matched
Pre-Condition	Finding corresponding features from two similar datasets based
Post Condition	when the feature matching is used to derive to transfer attributes

Table 3.7 Functional Requirement 07

Functional Requirement ID	**FR-07**
Functions	**Recognize Gesture**
Description	Gesture recognize from the dictionary
Input Source	Gesture recognize
Output Source	Gesture can be recognized
Pre-Condition	Gesture is defined any physical moment large or small that can be interpreted by a webcam.
Post Condition	Hand gesture is predicted according to the number and content of fingers detected.

Table 3.8 Functional Requirement 08

Functional Requirement ID	**FR-08**
Functions	**Display Result**
Description	Result can be displayed
Input Source	Recognize gesture of result
Output Source	Result displayed
Pre-Condition	System will make sure to take correct gesture in the class of gestures.
Post Condition	Result will show

3.6.2 Non-Functional Requirements (NFR)

Non-functional requirements specify the criteria in the operation and the architecture of the system.

Table 3.9 Non-Functional Requirement 01

NFR-01 Efficiency in Computation
This software shall minimize the use of Central Processing Unit (CPU) and memory resources on the operating system. When HGR (**Hand Gesture Recognition**) is executing, the software shall utilize less than 80% of the system's CPU resource and less than 100 megabytes of system memory.

Table 3.10 Non-Functional Requirement 02

NFR-02 Extensibility
The software shall be extensible to support future developments and add-ons to the HGR software. The gesture control module of HGR shall be at least 50% extensible to allow new gesture recognition features to be added to the system.

Table 3.11 Non-Functional Requirement 03

NFR-03 Portability
The HGR software shall be 100% portable to all operating platforms that support Java Runtime Environment (JRE). Therefore, this software should not depend on the different operating systems.

Table 3.12 Non-Functional Requirement 04

NFR-04 Performance
This software shall minimize the number of calculations needed to perform image processing and hand gesture detection. Each captured video frame shall be processed within 350 milliseconds to achieve 3 frames per second performance.

Table 3.13 Non-Functional Requirement 05

NFR-05 Reliability
The HGR software shall be operable in all lighting conditions. Regardless of the brightness level in user's operating environment, the program shall always detect user's hands.

Table 3.14 Non-Functional Requirement 06

NFR-06 Usability
This software shall be easy to use for all users with minimal instructions. 100% of the languages on the graphical user interface (GUI) shall be intuitive and understandable by non-technical users.

Table 3.15 Non-Functional Requirement 07

NFR-07 Accessibility
System should be easily accessible to every user.

Table 3.16 Non-Functional Requirement 08

NFR-08 Cost
Total building cost along with maintenance cost of software should be affordable.

Table 3.17 Non-Functional Requirement 09

NFR-09 Performance
Overall performance of the system should be very reliable and excellent.

Table 3.18 Non-Functional Requirement 10

NFR-10 Time complexity
There should be no time complexity in the system and system should work fast so that user can get the output within seconds.

Table 3.19 Non-Functional Requirement 11

NFR-11 Efficient
Overall performance and working of the system should be efficient.

3.7 Deployment Requirements

There are various requirements (Software and services) to successfully deploy the system. These are mentioned below:

3.7.1 Hardware Requirements

Hand gesture recognition software does not require special equipment except for a personal computer (PC) and a webcam. The CPU of this computer should have at least two cores to handle the enormous amount of calculations needed for the image processing unit.

3.7.2 Software Requirements

- OpenCV
- Python and its modules(libraries)

3.7.3 Non-Functional Requirements

- Memory Space
- Response Time

3.8 Hurdles in Optimizing the Current System

- Storage issues.
- Advance Python modules having change their attributes in every updation

3.9 Chapter Summary

Requirement Engineering - Starts with a project proposal based on need, after the benefits of the project. This chapter also explains the software engineering model used to represent different designs. The chapter also includes details of major project modules. The chapter also provides insights into the various studies conducted for the project. It later provides details of the various requirements for the development project.

CHAPTER 4

DESIGN

4.1 Software Process Model

The Cross Industry Standard Process for Data Mining **(CRISP-DM)** is a process model with six phases that naturally describes the data science life cycle. It's like a set of guardrails to help you plan, organize, and implement your data science (or machine learning) project.

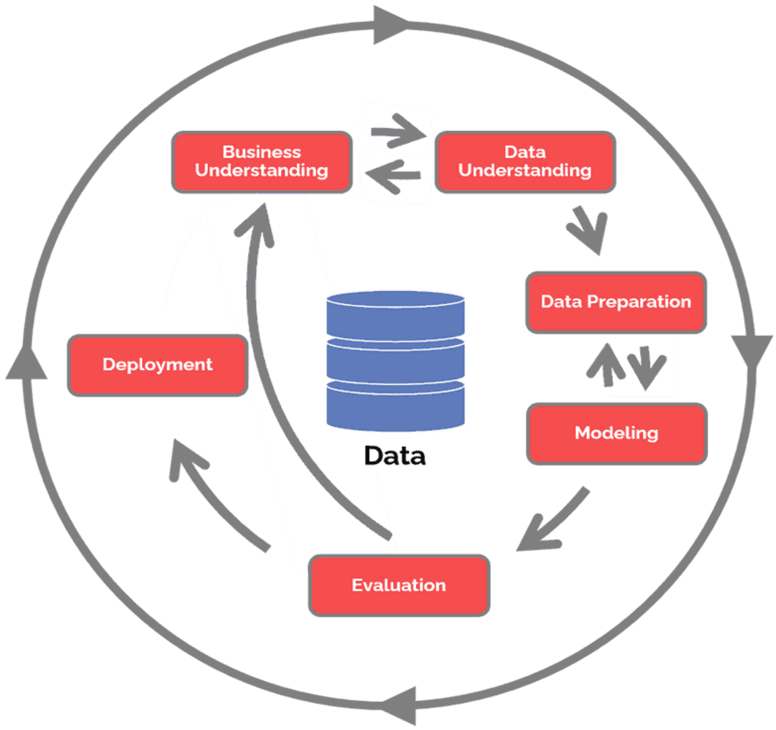

Figure 4.1 CRISP-DM Model

Published in 1999 to standardize data mining processes across industries, it has since become the most common methodology for data mining, analytics, and data science projects.

Data science teams that combine a loose implementation of CRISP-DM with overarching team-based agile project management approaches will likely see the best results. Even teams that don't explicitly follow CRISP-DM, can still use the framework diagram to explain how the differences between data science and software projects.

4.1.1 Benefits of Model

Table 4.1 Benefits of Model

Features	Description
Generalize-able	CRISP-DM provides strong guidance for even the most advanced of today's data science activities.
Common Sense	When students were asked to do a data science project without project management direction, they "tended toward a CRISP-like methodology and identified the phases and did several iterations."
Adopt-able	Like Kanban, CRISP-DM can be implemented without much training, organizational role changes, or controversy.
Right Start	The initial focus on Business Understanding is helpful to align technical work with business needs and to steer data scientists away from jumping into a problem without properly understanding business objectives.
Strong Finish	Its final step Deployment likewise addresses important considerations to close out the project and transition to maintenance and operations.
Flexible	A loose CRISP-DM implementation can be flexible to provide many of the benefits of agile principles and practices.
Feedback	It's easy to get customer feedback on developments already happening. Users can comment on demonstrations of the software and see how well it is implemented. Users find it difficult to decide progress with the help of software design documents.

4.1.2 Limitations of Model

- The work of this project will be a bit slow as the deep learning of the framework requires advanced processing hardware and graphical processing unit system.
- The models we use to identify faces are pre-trained models. Therefore, if we want to train our model, it requires a lot of time and processing.
- In the system, scanning of each frame is 3-4 per second but still needs to be improved. If faces move too fast, they cannot be detected.
- Management is more complex.
- Process is complex.
- The end of the project will not be known soon.

4.2 Phases and Tasks

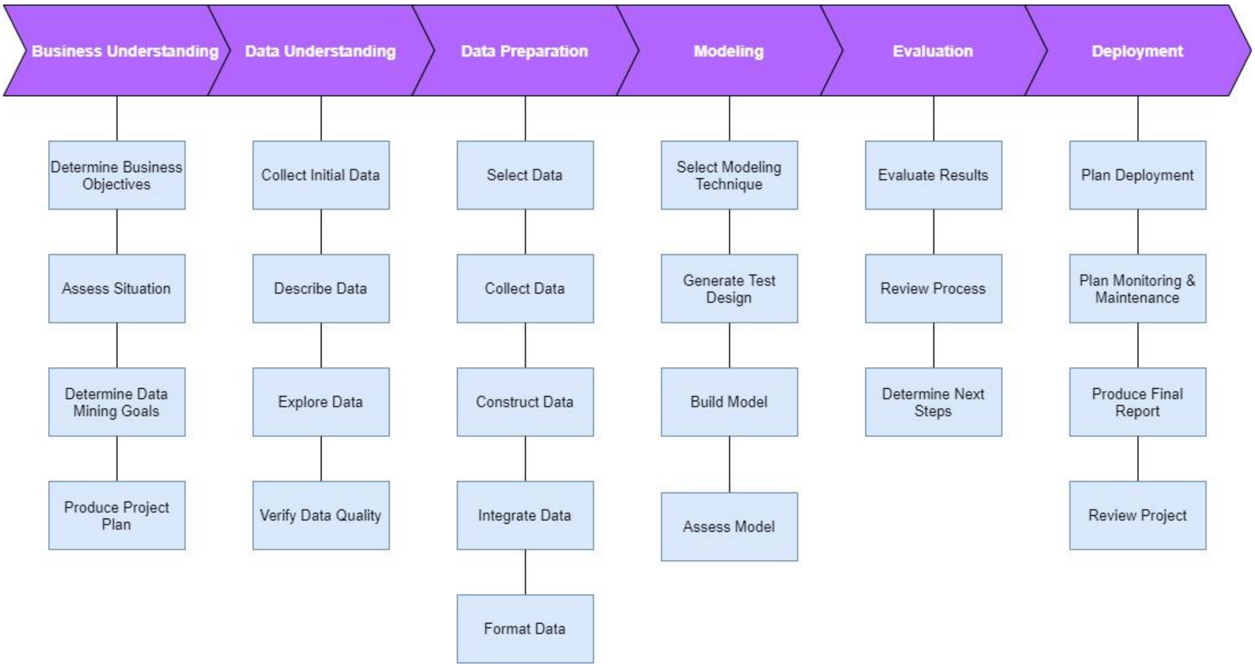

Figure 4.2 Phases and Tasks

4.3 Characteristics of CRISP-DM

- It encourages data miners to focus on business goals, so as to ensure that project outputs provide tangible benefits to the organization.
- The CRISP-DM approach helps ensure that the business goals remain at the center of the project throughout.
- CRISP-DM provides an iterative approach, including frequent opportunities to evaluate the progress of the project against its original objectives.

- It also means that the project stakeholders can adapt & change the objectives in the light of new findings.
- The CRISP-DM methodology is both technology and problem-neutral.
- Whatever the nature of your data mining project, CRISP-DM will still provide you with a framework with enough structure to be useful.

4.4 Design

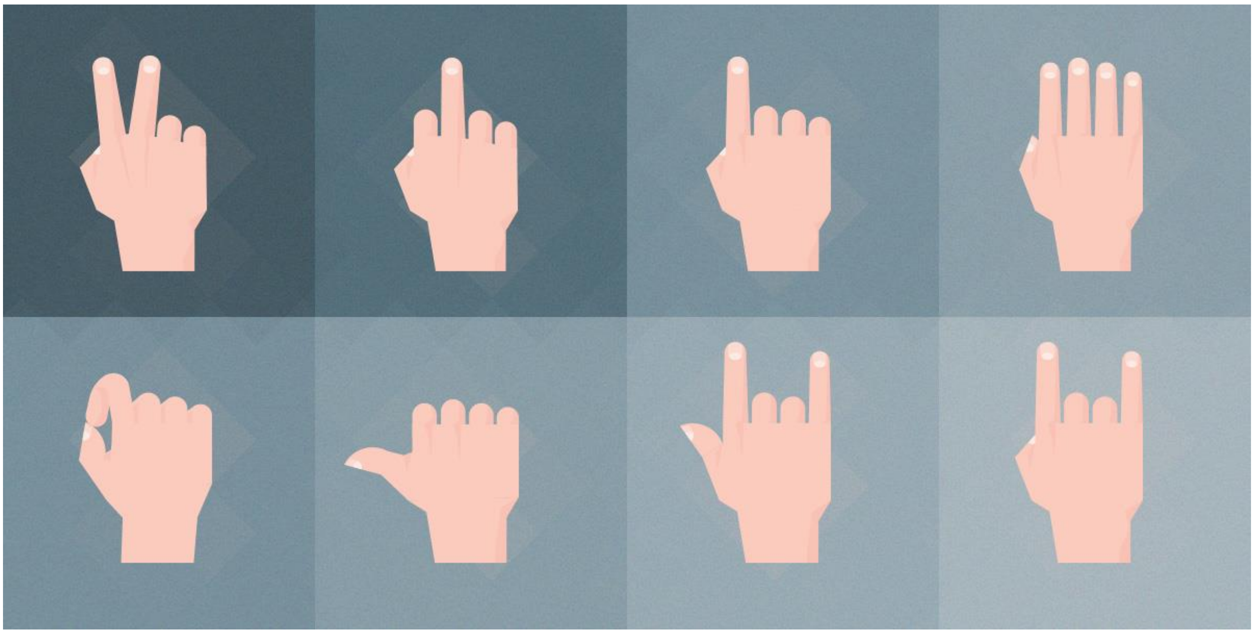

Figure 4.3 Design of hand gesture recognition system

4.4.1 Description

This diagram shows that the gesture of hand mostly used in hand gesture recognition system with the help of these gesture we can control many things in our system. All five finger will be used for some functionality in our system we can control our system with our hand gesture.

4.4.2 Methodology of the Proposed System

Hand gesture recognition project is done using Python programming language and OpenCV as library. Python programming language produces simple and easy system code to understand. Also, Python package used here is Numpy. The image that is captured using web camera will be processed in a region called as Region of Interest (ROI) where act as a region of wanted area while ignoring the outside region, called background.

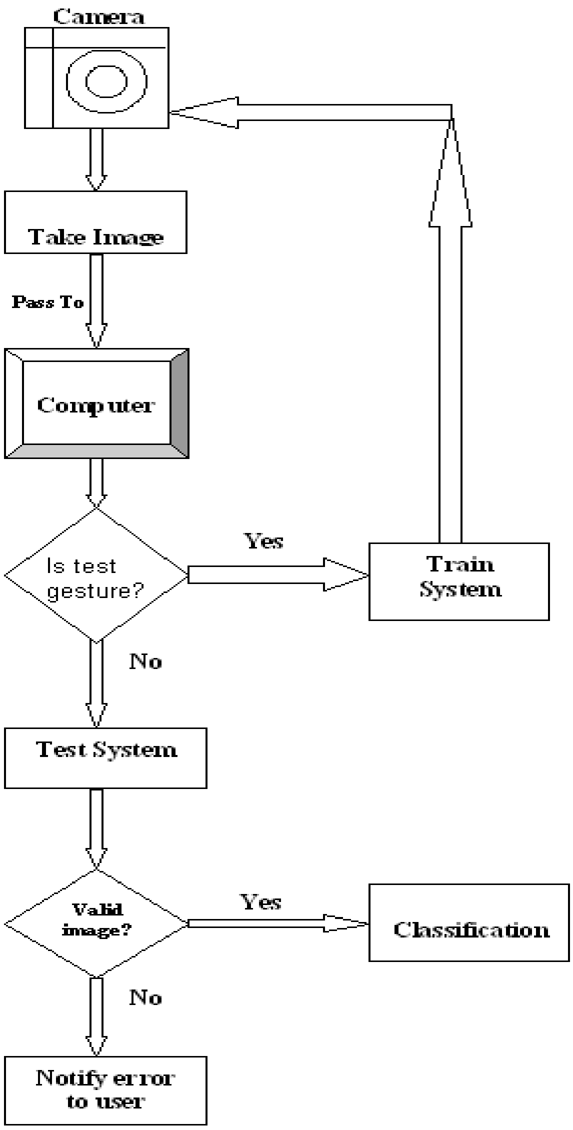

Figure 4.4 Methodology Diagram

4.4.3 RGB to Grayscale

RGB stands for Red, Green and blue. It is a system of colors in which these three mentioned colors are added in different quantities to give different colors. A human's ability of visions can distinguish between many different colors, their intensities and shades. When it comes to the shades of gray, human vision can only distinguish approximately 100 shades of gray. So it is evident from this fact that the images that colored contain more information. RGB to Grayscale Gray filtering using value Binarize Noise removal and smoothing Remove small objects other than hand Region filling.

4.4.4 Grayscale filtering using value

There are many different type of filters in the field of Digital Image Processing, Gray level filter is one of them. This filter works on gray level image. The aim is to reduce noise in order to increase accuracy and get better results out of this system. In this a threshold is used to filter out noise in grayscale image. The threshold used in this project was 75, it was giving better results.

4.4.5 Binarize

Binarization is a process which converts a gray level image to a binary image. Gray level image has 0 to 255 levels, whereas in binary image there are only two values; 0 and 1(black and white).

4.4.6 Noise removal and smoothing

Noise is actually a variation in an image or unwanted and undesired changes in the color or brightness of an image. Noise in the image need to be removed, because it will affect the results. If features extracted from a noisy image are used and then it is classified, it will be misleading and will result in bad classification and results, so in-order to avoid this image is preprocessed by removing noise from this image. It will increase the accuracy of the system.

4.4.7 Removing small objects other than hand

The object of interest is the hand, not other small objects or noise acting as a small object in the images. This biggest object in this case which is hand is called biggest BLOB. In this step a threshold of 50 was used, that removed all the connected components that have a pixel size lower than 50, it means remove all the objects that have pixels smaller than 50. As a result, only the biggest object is extracted, which is hand in this case.

4.4.8 Classification

Classification involves two basic steps

4.4.8.1 Machine Learning

4.4.8.2 Recognition

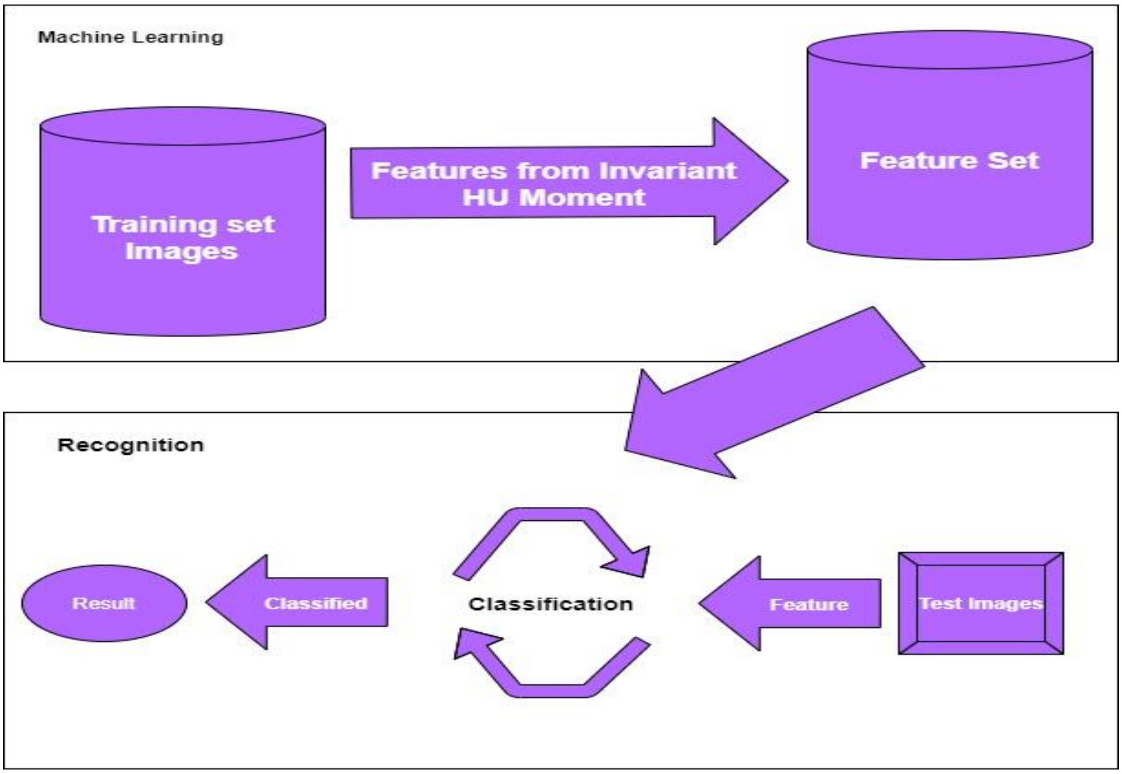

Figure 4.5 Machine Learning and Recognition

4.4.9 UML Diagrams

The UML diagram is a diagram based on the UML (Unified Modelling Language) that aims to represent a system's main actors, roles, functions, patterns, or classes, in order to better understand, change, retain, or document Information can be obtained.

There are several types of UML diagrams. Some of which are as follows:

4.4.9.1 Use Case Diagram

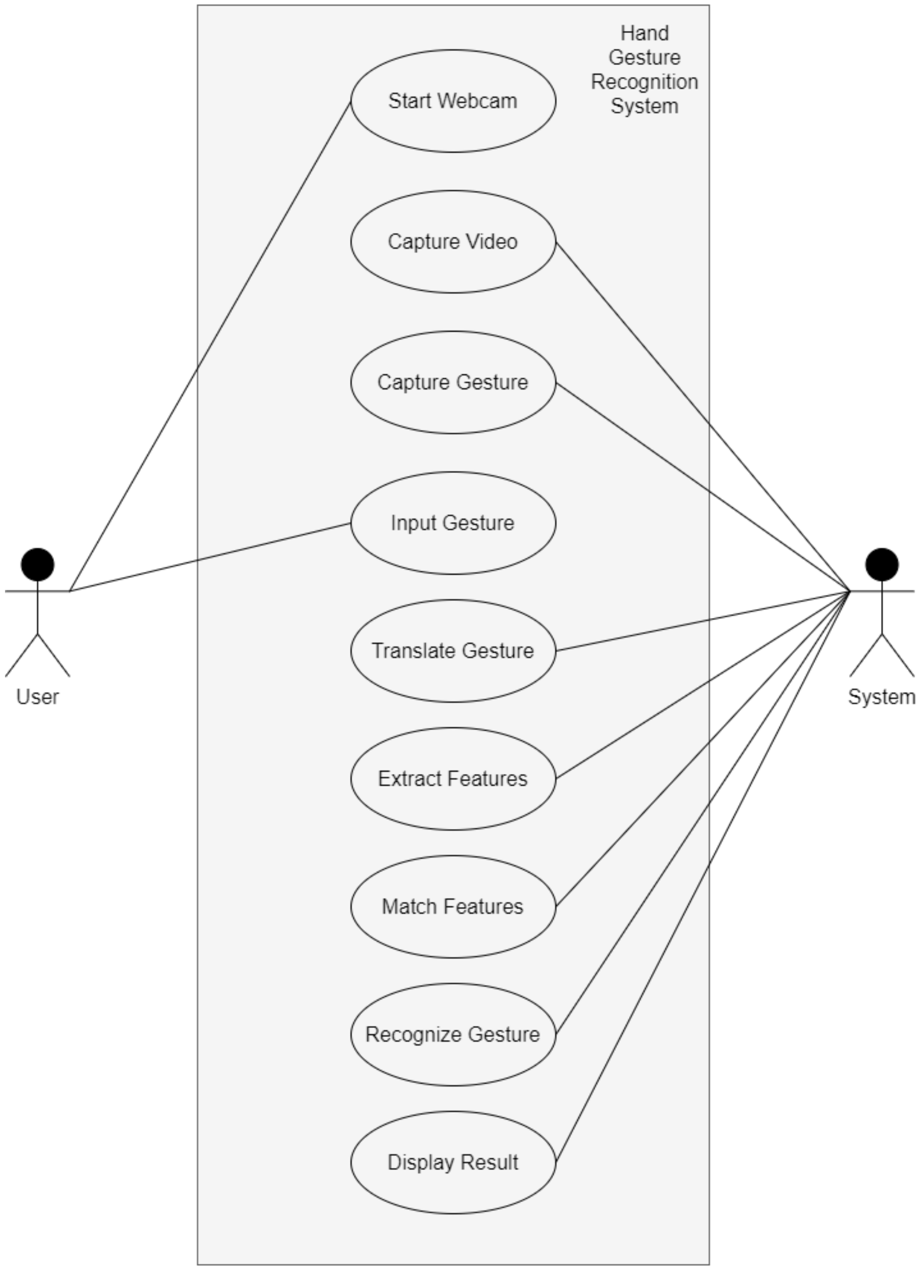

Figure 4.6 Use Case Diagram

4.4.9.2 Sequence Diagram

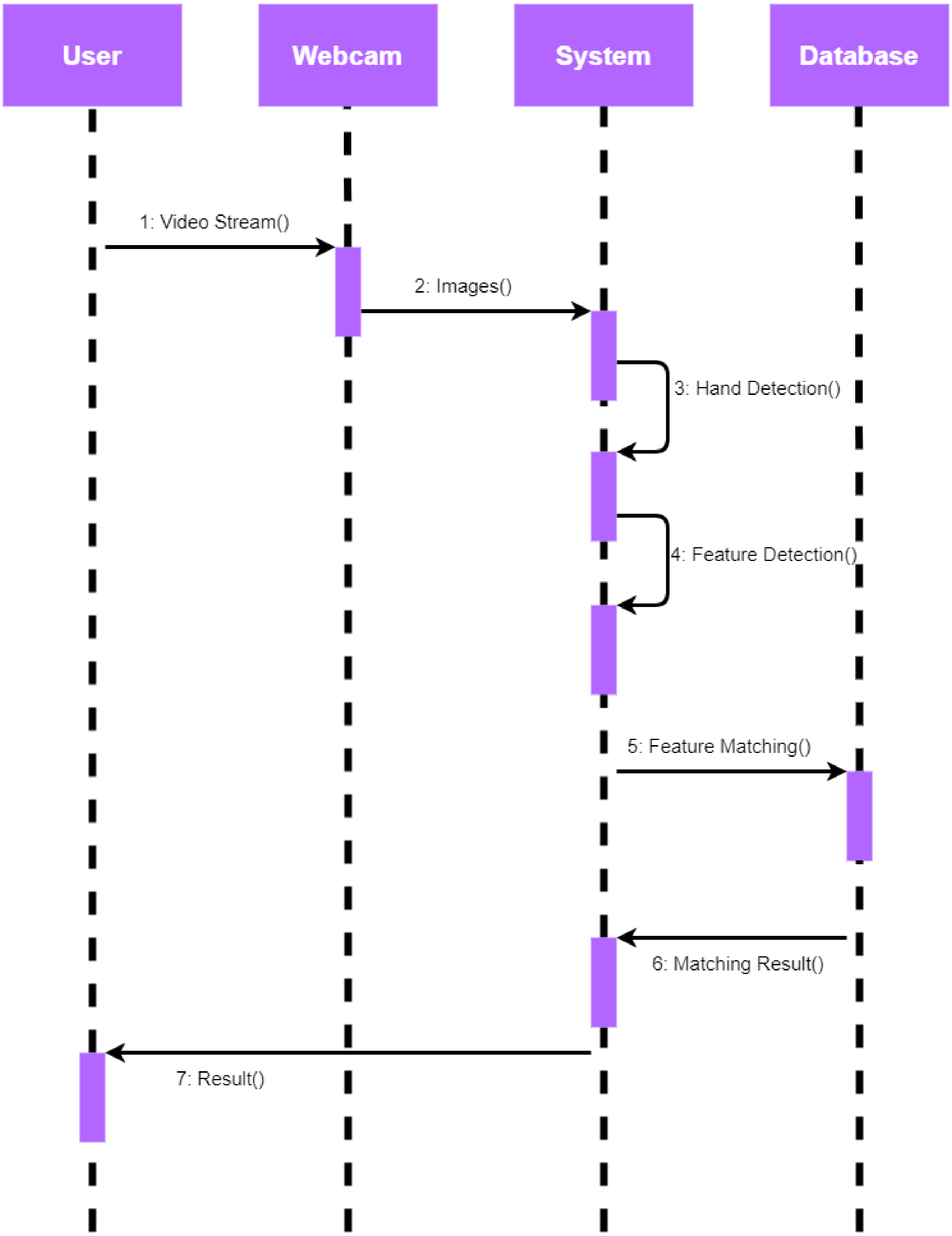

Figure 4.7 Sequence Diagram

4.4.9.3 Activity Diagram

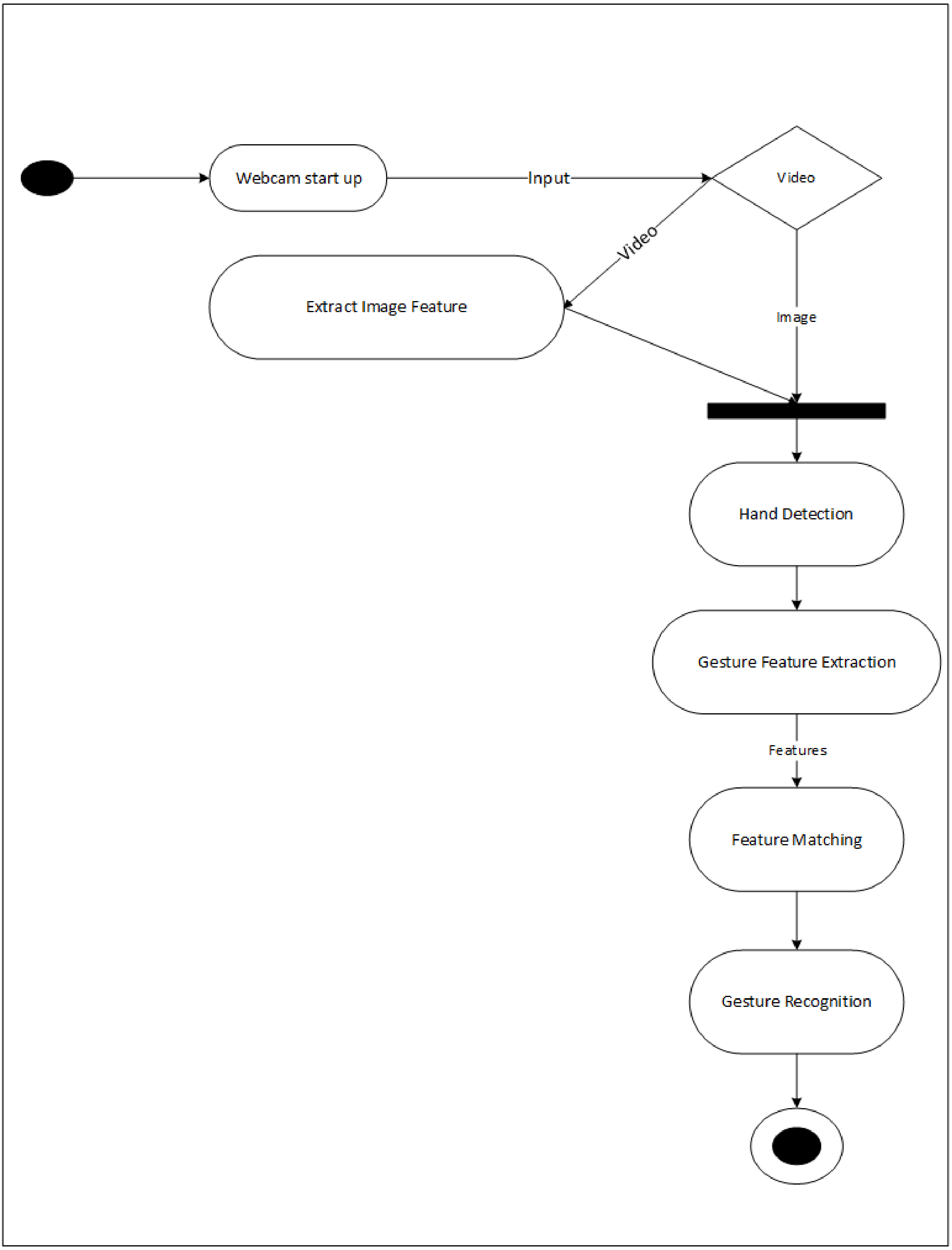

Figure 4.8 Activity Diagram

4.5 Chapter Summary

In this chapter we discussed the process model which we are using to develop this hand gesture recognition system which is Incremental Model. We discussed the benefits for our system to use the incremental model which mainly includes that we bring betterment after feedback from user in increment. We also described the limitation for the Incremental Mode. Further there are UML diagram that are used to visually represent a system with its main specific actors, actions, roles to better understand, maintain or document information about the system.

CHAPTER 5

DEVELOPMENT AND IMPLEMENTATION

5.1 Development of Computer Program

The process of designing and developing an executable computer programe to complete a specified computing job is known as computer programming. Programming includes activities such as analysis, algorithm generation, algorithm accuracy and resource consumption assessment, and algorithm implementation in a programming language of choice. A program's source code is written in one or more programming languages.

5.2 Tool Selection

Hand Gesture Recognition System is developed using python.

5.2.1 Hardware

We use hardware only the webcam of Computer System.

5.2.2 Software

- Python 3.9 version for 64-bit system.
- PyCharm

5.2.3 Platform Selection

PyCharm IDE and python scripting language were used to develop the code of Hand Gesture Recognition System.

5.3 Program Coding

```
import cv2
import mediapipe as mp
import pyautogui
import win32api
```

Figure 5.1 Importing Modules

```
mp_drawing = mp.solutions.drawing_utils
mp_hands = mp.solutions.hands
```

Figure 5.2 Hand Tracking Modules

```
def fingerPosition(images, handNo=0):
    lmLists = []
    if results.multi_hand_landmarks:
        myHand = results.multi_hand_landmarks[handNo]
        for ids, lm in enumerate(myHand.landmark):
            # print(id,lm)
            # h is height w is width and c is channel
            h, w, c = images.shape
            # find the position of the center decimal to int convert
            cx, cy = int(lm.x * w), int(lm.y * h)
            lmLists.append([id, cx, cy])
    return lmLists
```

Figure 5.3 Coordinates of Finger Tips

```
cap = cv2.VideoCapture(0)
cap.set(3, wCam)
cap.set(4, hCam)
```

Figure 5.4 Video Streaming with Webcam

```
with mp_hands.Hands(
        min_detection_confidence=0.8,
        min_tracking_confidence=0.7) as hands:
    while cap.isOpened():
        success, image = cap.read()
```

Figure 5.5 Detection & Tracking Performance

```
image = cv2.cvtColor(cv2.flip(image, 1), cv2.COLOR_BGR2RGB)
image.flags.writeable = False
results = hands.process(image)
# Draw the hand annotations on the image.
image.flags.writeable = True
image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
```

Figure 5.6 Hand Detection

```
if results.multi_hand_landmarks:
    for hand_landmarks in results.multi_hand_landmarks:
        mp_drawing.draw_landmarks(
            image, hand_landmarks, mp_hands.HAND_CONNECTIONS)
lmList = fingerPosition(image)
```

Figure 5.7 Drawing Landmarks

```
if results.multi_hand_landmarks != None:
    for handLandmarks in results.multi_hand_landmarks:
        for point in mp_hands.HandLandmark:

            normalizedLandmark = handLandmarks.landmark[point]
            pixelCoordinatesLandmark = mp_drawing._normalized_to_pixel_coordinates(normalizedLandmark.x,
                                                                                   normalizedLandmark.y, wCam, hCam)

            point = str(point)
```

Figure 5.8 Normalize Pixel Coordinates

```
if point == 'HandLandmark.INDEX_FINGER_TIP':
    try:
        indexfingertip_x = pixelCoordinatesLandmark[0]
        indexfingertip_y = pixelCoordinatesLandmark[1]
        win32api.SetCursorPos((indexfingertip_x * 3, indexfingertip_y * 3))

    except:
        pass
```

Figure 5.9 Cursor Movement

```
if len(lmList) != 0:
    fingers = []
    for id in range(1, 5):
        if lmList[tipIds[id]][2] < lmList[tipIds[id] - 2][2]:
            # print("open")
            fingers.append(1)
        if lmList[tipIds[id]][2] > lmList[tipIds[id] - 2][2]:
            fingers.append(0)
```

Figure 5.10 Fist Functionality

```
totalFingers = fingers.count(1)

if totalFingers == 4:
    state = "play"

if totalFingers == 2 and state == "play":
    state = "pause"
    pyautogui.click()
    print("Click")

if totalFingers == 3 and state == "play":
    state = "pause"
    pyautogui.click(button='right')
    print("Right options")

if totalFingers == 1:
    if lmList[8][1] < 300:
        print("left")
        pyautogui.press('left')
    if lmList[8][1] > 400:
        print("Right")
        pyautogui.press('Right')

if totalFingers == 0:
    if lmList[8][2] < 210:
        print("Up")
        pyautogui.press('Up')
    if lmList[8][2] > 230:
        print("Down")
        pyautogui.press('Down')
```

```
if totalFingers == 0 and state == "play":
    state = "pause"
    pyautogui.press('space')
    print("space")
```

Figure 5.11 Basic Functionality

```
    cv2.imshow("Media Controller, Game Controller & Mouse Trace", image)
    key = cv2.waitKey(1) & 0xFF
    # if the `s` key was pressed, break from the loop
    if key == ord("s"):
        break
cv2.destroyAllWindows()
```

Figure 5.12 Code Break

5.4 Chapter Summary

One platform is chosen, PyCharm for coding. Selection of tools and platform are decided after seeing compatibility. After choosing platform we code in it and did our best to meet requirements of our project.

CHAPTER 6

TESTING

6.1 Testing

The software testing is a critical element of software quality assurance. It represents the ultimate review of the specification, design, and coding. The main objective of the system testing is to find the errors or bugs in the system and to see whether the system fulfils expectations of the user. Testing is a procedure of executing a program with the intent of finding errors. But testing cannot show the absence of defects. It can show only that software errors are present.

6.1.1 System Testing

The system testing is the process to evaluate whether the system has fulfilled 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.

- To affirm the quality of the project
- To find and eliminate any residual errors from previous stages.
- To validate the software as a solution to the original problem.
- To provide the operational reliability of the system.

6.1.2 Quality Assurance

Quality assurance consists of the auditing and reporting functions of management. The goal of quality assurance in hand gesture recognition system is to provide management with the data necessary to be informed about product quality, thereby gaining insight and confident that the product quality is meeting its goals. This is an “umbrella activity” that is applied throughout the engineering process. Software quality assurance encompasses the number of fingers raised.

- Formal technical reviews that are applied during each software engineering.
- Feature such as contour and convex hull to be extracted without any distortion.
- Control of software documentation and the change made to it.
- A procedure to ensure compliance with software development standards.
- Measurements and reporting mechanisms of different pre-processed.

6.1.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Table 6.1 Functional Test

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	Color space conversion	Display GBR image in new window	GBR image is displayed in new window	Pass
4	Image filtering	Reduce noise in the image	Noise is reduced	Pass
5	Noise is reduced	Display hand region in white with black background	Hand region is displayed in white without background	Pass
6	Features extraction	Display of hand center	Hand center is displayed	Pass
		Hand center is displayed	Palm radius is drawn	Pass
		Display of fingertips	Fingertips is displayed	Pass
7	Gesture recognition	Display “Play” when show full palm gesture	“Play” is displayed	Pass
8	Gesture recognition	Display “Pause” when show fitch closed gesture	“Pause” is displayed with slightly delay	Pass
9	Gesture recognition	“Mouse move” when show index finger tip gesture	“Mouse Movement” Through Index Finger Tip	Pass

10	Gesture recognition	Display “Click” when show index finger and middle finger	“Click” is displayed	Pass
11	Gesture recognition	Display “Right Click” when show index, middle and ring Fingers	“Right Click” is displayed	Pass
12	Gesture recognition	Display “Right Movement” when show index finger move right side	“Right Movement” is displayed	Pass
13	Gesture recognition	Display “Left Movement” when show index finger move left side	“Left Movement” is displayed	Pass
14	Gesture recognition	Display “Up” when show closed fitch move up	“Up Movement” is displayed	Pass
15	Gesture recognition	Display “Down” when show closed fitch move Down	“Down Movement” is displayed	Pass
16	Quit Program	Exit when “q” button is pressed	All window is terminated	Pass

6.2 Chapter Summary

In testing phase, we find many difficulties and even minor negligence leads us to faulty working. We face image tracking and gesture issue. We recheck everything and code again and again until and unless we get our required tasks.

REFERENCES

[1]. J.Jenkinwinston (96207106036), M.Maria Gnanam (96207106056), R.Ramasamy (96207106306), Anna University of Technology, Tirunelveli: Hand Gesture RecognitionsystemUsing Using Haar Wavelet. [https://www.scribd.com/document/66477076/Hand-Gesture-Recognition-Using-Haar-wavelet]

[2]. Laura Dipietro, Angelo M. Sabatini, Senior Member, IEEE, and Paolo Dario, Fellow, IEEE, A Survey of Glove-Based Systems and Their Applications. [https://ieeexplore.ieee.org/document/4539650?denied=]

[3]. Kay M. Stanney HANDBOOK OF VIRTUAL ENVIRONMENTS Design, Implementation, and Applications, Gesture Recognition Chapter #10 by Matthew Turk [https://www.routledge.com/Handbook-of-Virtual-Environments-Design-Implementation-and-Applications/Hale-Stanney/p/book/9781138074637]

[4]. Daniel Thalman, Gesture Recognition Motion Capture, Motion Retargeting, and Action Recognition [https://www.google.com/search?q=Daniel+Thalman%2C+Gesture+Recognition+Motion+Capture%2C+Motion+Retargeting%2C+and+Action+Recognition&oq=Daniel+Thalman%2C+Gesture+Recognition+Motion+Capture%2C+Motion+Retargeting%2C+and+Action+Recognition&aqs=chrome.69i57.1702j0j7&sourceid=chrome&ie=UTF-8]

[5]. Hatice Gunes, Massimo Piccardi, Tony Ja, 2007, Research School of Information Sciences and Engineering Australian National University Face and Body Gesture Recognition for a Vision-Based Multimodal Analyzer multimodal [https://scholar.google.dk/citations?user=2WkE1wYAAAAJ&hl=ja]

[6]. J. Heinzm ann and A. Zelinsky Robust Real - Time Face Tracking and Gesture Recognition [http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.106.9709]

[7]. Vladimir Vezhnevets, Vassili Sazonov, Alla Andreeva, Moscow State University A Survey on Pixel-Based Skin Color Detection Techniques, [https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.189.5454&rep=rep1&type=pdf]

[8]. TEY YI CHI, Universiti Teknologi Malaysi, FUZZY SKIN DETECTION [https://scholar.google.com.pk/scholar?q=TEY+YI+CHI,+Universiti+Teknologi+Malaysia,+FUZZY+SKIN+DETECTION&hl=en&as_sdt=0&as_vis=1&oi=scholart]

[9]. Ray Lockton, Balliol College, Oxford University, Hand Gesture Recognition Using Computer Vision.[https://www.yumpu.com/en/document/read/5329170/4-year-project-report-hand-gesture-recognition-using-computer-]

[10]. Howard Demuth, Mark Beale, Neural Network Toolbox [http://cda.psych.uiuc.edu/matlab_pdf/nnet.pdf]

[11]. Ari Y. Benbasat and Joseph A. Paradiso in MIT Media Laboratory, Cambridge An Inertial Measurement Framework for Gesture Recognition and Applications [https://www.researchgate.net/publication/221260354_An_Inertial_Measurement_Framework_for_Gesture_Recognition_and_Applications] Peggy Gerardin, Color Imaging Course Winter Semester 2002-2003, Color Image Segmentation [https://scholar.google.fr/citations?user=RuXpiMAAAAAJ&hl=fr.

[12]. S. Padmappriya & K. Sumalatha – Digital Image Processing Real Time Application;International Journal Of Engineering Science Invention(IJESI),2018

[13]. OpenCV python hand gesture recognition. http://creat-tabu.blogspot.com/2013/08/opencv-python-hand- gesture-recognition.html [Accessed 7 August 2018]