A Final Year Project

On

Hand Gesture Recognition System

Under the Supervision of

Mr. Bikash Balami

Asian School of Management and Technology

Gongabu, Kathmandu

Submitted By

Bishal Bhandari [5532/071]

Kshitiz Sapkota [5546/071]

Pralhad Rijal [5550/071]

Prem Lamsal [5551/071]

B.Sc.CSIT

Department of Computer Science and Information Technology

Asian School of Management and Technology

Gongabu, Kathmandu

Submitted To

Institute of Science and Technology

Tribhuvan University Kritipur, Kathmandu

ACKNOWLEDGEMENT

We owe our most profound appreciation to the Asian School of Management and Technology for furnishing us with a chance to work on project as part of our syllabus. We are generously obligated to our supervisor Mr. Bikash Balami for his consistent help and direction all through this project. It was his significant recommendation and suggestion that helped us to adapt up to the rising impediments amid the improvement of this project.

We are additionally thankful to Mr. Surya Bam, for the underlying guidance in our project and spurred us to put more exertion in this project with providing a flexible working environment. We are likewise thankful to every one of our respected educators for their motivation and encouragement that made ready towards the completion of this project.

We would like to enhance our sincere regards to our respected principal Er. Anil Lal Amatya and all the faculty members who help us directly or indirectly for completing this project. We would also like to thank all of our friends, senior and colleague for their valuable suggestion, comments and support. Also, we are grateful to Mr. Ashok Kumar Pant for his immense support to our project.

Thanking You,

Bishal Bhandari (Roll No. 5532/071)

Kshitiz Sapkota (Roll No. 5546/071)

Pralhad Rijal (Roll No. 5550/071)

Prem Lamsal (Roll No. 5551/071)

ABSTRACT

Human centered human-computer interaction technology tends to replace the traditional computer centered technology with its growing applicability to a wide variety of applications. Acting as a way of most natural communication between human and machine, vision-based hand gesture recognition is becoming the pursuit of human-computer interaction. General vision-based hand gesture recognition generally consists of sample capturing, image preprocessing, feature extraction and classification. Among these procedures feature extraction aims to detect and extract features that can be used to determine the meaning of a given hand gesture. The extracted features should be able to describe gesture uniquely and be robust to the shift and rotation of hand gesture in order to achieve a reliable recognition. In this paper, we propose a method to extract a series of features based on convex defect detection, taking advantage of the close relationship of convex defect and fingertips.

When we use gesture, we enter into a whole history of human communication because there is no language that exists entirely without gesture. People cannot communicate without gesture. It is so connected to the intention that there is a phrase "empty gesture" used to mean an action or movement that is without genuine feeling. The research centralizes on the efforts of implementing an application that employs image processing methods and gesture recognition techniques, which in turn results into the development of a system that interacts with environment.

Keywords:

Gesture Recognition, Human-Computer Interaction, Real time based hand Gesture

Table of contents

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

3D (3-Dimension)

AI (Artificial Intelligence)

HCI (Human Computer Interaction)

HSV (Hue Saturation Value)

IDE (Interactive Development Environment)

ML (Machine Learning)

OpenCV (Open Source Computer Vision)

CHAPTER ONE

INTRODUCTION

1.1. Introduction

Recent developments in computer software and related hardware technology have provided a value-added service to the users. In everyday life, physical gestures are a powerful means of communication. They can economically convey a rich set of facts and feelings. Gesture recognition is a topic in computer science and language technology with the goal of interpreting human gestures via mathematical algorithms. Gestures can originate from any bodily motion or state but commonly originate from the face or hand. With the arts and science of technology and computing, computer is becoming greater and preferably suited in our past today life. Various input and produce devices have been designed and used from a well-known end to the other the forever and ever the final cause of easing the communication surrounded by automation and humans [14]. In circumstance, more and preferably unrestricted interaction mid humans and computers are established in personal digital assistant applications. The idea behind it is to impress mechanics in a style so that computers can understand human explanation and shake a freak friendly human personal digital assistant Interactions. Making a computer understand human cross is a lead towards it. A sign of the cross is spatiotemporal knee-jerk reaction, which commit be aspersion or forceful or both Bobick and Wilson most zoned gestures as the motion of the advantage to communicate mutually other agents. In recent ages, cross letter of recommendation route has added very popular in the employment of delve in to, by style of explanation facial and hand gesture passport system [11].

The task of hand gesture recognition is one the important and elemental problem in computer vision. With recent advances in information technology and media, automated human interactions systems are built which involve hand processing task like hand detection, hand recognition and hand tracking. Hand gestures bouncecel be covert in two categories: animadversion and dynamic. Whereas static sign of the cross is a particular member of the working-class configuration and fake, represented by a single thought, a bold cross is a moving gesture, represented by a solution of images. Sign definition is one of the approximately structured fit of gestures. In authenticate definition, each gesture has a tenacious meaning. Hand gestures provide a more human-computer interface, allowing us to point, or swivel a 3D ideal by rotating hands. One of the most suited applications of laborer gesture recognition is the easy way communication by the whole of the hard of hearing or non-vocal persons through a hand-gesture to style system to enliven the quality of life. Deaf or non-vocal individuals are weak to communicate by all of others over speech what is coming to one to congenital malfunction, epidemic, chief injuries. Deaf or non-vocal persons use authenticate language or common laborer gestures to lay it on the line themselves. However, practically of the tribe who have hearing gift do not have the in a class by itself sign language expertise. This is a major barrier between these two groups in by the day communication. To overcome this stone in one path and to throw in one lot with those heirs and assign to integrate into community is a literally challenging delve in to area.

This prompted our interest so we planned to make a software system that could recognize human hand gestures through computer vision, which is a sub field of artificial intelligence. The purpose of my software through computer vision was to program a computer to "understand" a gesture or features in an image.

1.1.1. Motivation

In today's world, the need of maintaining the security of information and physical property has become increasingly important and difficult. If we are able to count finger(s) of person automatically, then this simple test results in the development of a more intelligent system that can be commanded by the gestures. So, we were interested to develop a system that can recognize hand gesture. Our project study provides a better, deeper and a clearer understanding of overall process involved in transforming the static image into recognizable gesture by the computer device.

1.2. Problem Definition

The conventional way of identifying and responding has few limitations in term of accuracy. As we do research and study there are many specific topics in the field of image processing and gesture recognition. With the development of ubiquitous computing, current user interaction approaches with keyboard, mouse and pen are not sufficient. Due to the limitation of these devices the usable command set is also limited. Direct use of hands can be used as an input device and recognition for providing natural interaction. Human Gesture Recognition is designed in order to cater the need of fast growing world to minimize the hardware caring burden.

Recognizing hand gesture has really been a tough task and is creating several problems and also several forgery cases have occurred due to lack of good system for verification of hand gesture. Deaf or non-vocal individuals are weak to communicate by all of others over speech what is coming to one to congenital malfunction, epidemic, chief injuries. To overcome this problem a good hand gesture recognition system is necessary to developed.

1.3. Objectives

The objective of this study is to detect, recognize, interpret and implement the Hand Gesture Recognition System using Convex Hull and Contour. And, hence to build a model, that will recognize the hand gesture into recognized numbers. The main objective is given below,

1. To recognize and translate the hand gesture into accurate Number.
2. Successfully implement a method to extract useful features.

1.4. Scopes and limitations

The scope for this project work is to upgrade the human-computer interaction. The main focus of the project work "Hand Gestures recognition" is to classify multiple classes of number through hand gestures. This project uses the input devices to feed the data by using different hand gesture class. We can enhance this system for the use of controlling the robots, communication etc. Some of the scopes are:

1. This system can be used as a security tool in home and offices.
2. The system can be used to recognition to counting system.
3. Developing aids for the deaf and mute people.
4. Enabling very young children to interact with a computer.
5. Recognizing sign language.
6. Navigating and/or manipulating virtual environments.
7. Communicating in video conferencing.
8. Distance learning / teleteaching assistance.
9. Graphics editor control.

The limitation of this project is that only the certain numbers can be recognized. The other gesture like directions, face detection is not recognized. The insufficient light can cause problem for the recognition of finger patterns and also if there are lots more noises in the picture then the system will not recognize the gesture or pattern of hand.

1.5. Report Organization

Our project starts with the hand gesture patterns from class one to five. Then we will feed it into the system where, features from those data will be extracted. In further, those features will be used to classify the class of given input gesture class.

The rest of the report is organized as follows: Chapter 2 reviews related works, requirement analysis and feasibility analysis of our project. Chapter 3 specify the system design of the project and describe the basic process model. Chapter 4 deals with the implementation and testing of the system. . Conclusions is discussed in Chapter 5.

CHAPTER TWO

REQUIREMENT AND FEASIBILITY ANALYSIS

2.1. Literature Review

The purpose of this project study is to recognize hand gestures. In this chapter, the major ground work and preliminaries related to the subject of the study, is review. The various related approaches and review of our project is review in his chapter.

2.1.1. Calculation using Gaussian (DOG)

In [1], the author includes the algorithm in which first the video was captured and then divided into various frames and the frame with the image was extracted and further from that frames various features like Difference of Gaussian (DOG) was calculated. This architecture considers the hierarchical composite properties of hand gestures and combines the advantages to achieve real-time vision-based hand tracking and gesture recognition.

2.1.2. Multidimensional hand model-based approach

In [2], the author applied a combination of (2D) shape-based and size-based features to recognize the configuration of the hand in the scene. The author gathered the tracking data over time with a 3D-camera, and the author devises SVM-based system to classify between one finger, two finger gestures. This three-dimensional hand model is based on the 3D kinematic hand model with considerable DOF's, and try to estimate the hand parameters by comparison between the input images and the possible 2D appearance projected by the 3D hand model. This approach is ideal for realistic interactions in virtual environments.

2.1.3. Implementation of HMM

In [5], the authors implemented Hidden Markov models, commonly used in handwriting recognition, achieve a 97%+ accuracy in classifying 40 words in American Sign Language. Hidden Markov Models (HMM) model deals with the dynamic aspects of gestures. Gestures are extracted from a sequence of video images by tracking the skin color blobs corresponding to the hand into a body– face space centered on the face of the user. The goal is to recognize two classes of gestures: deictic and symbolic. The image is filtered using a fast lookup indexing table. After filtering, skin color pixels are gathered into blobs. Blobs are statistical objects based on the location (x, y) and the colorimetry (Y, U, V) of the skin color pixels in order to determine homogeneous areas.

2.1.4. Hand Gesture Recognition Based on Convex Defect Detection

In [5], A method had been developed in which gesture recognition model was developed using convex defects and contour tracking. The authors have introduced the space for convex hull then use contour to find the area ratio in comparison to the total area. Current solutions use the real time technique to classify hands. The author had used a “δ” as a feature which represent the ratio contour and convex hull area.

2.1.5. Communication channels

There are essentially three types of communication channels available: audio, visual and environmental. Environment channels consist of interactions with the surrounding world, while audio cues are those that can be heard and visual cues those that can be seen. Depending on the technology medium used communication cues may, or may not, be effectively transmitted between the collaborators [6]. Research into human-robot interaction, the use of robots as tools, robots as guides and assistants, as well as the progress being made in the development of humanoid robots, are all examined [7]. Finally, a variety of efforts to use robots in collaboration are examined and analyzed in the context of the human-human model presented.

2.1.6. Data glove-based System

In [8], full American Sign Language recognition system (word, phrases) incorporate data gloves. *Takashi and Kishino* discuss a Data glove-based system that could recognize 34 of the 46 Japanese gesture using a joint angle and hand-oriented coding technique. From their paper, it seems the test user made each of the 46 gestures 10 times to provide data for principle component and cluster analysis.

2.2. Requirement Analysis

Requirement analysis is mainly categorized into two types:

2.2.1. Functional Requirements

A functional requirement is something a system must do. It includes the functions performed by specific screens, outlines of work flows performed by the system and other business of compliance requirements the system must meet.

2.2.1.1. Use Case Diagram

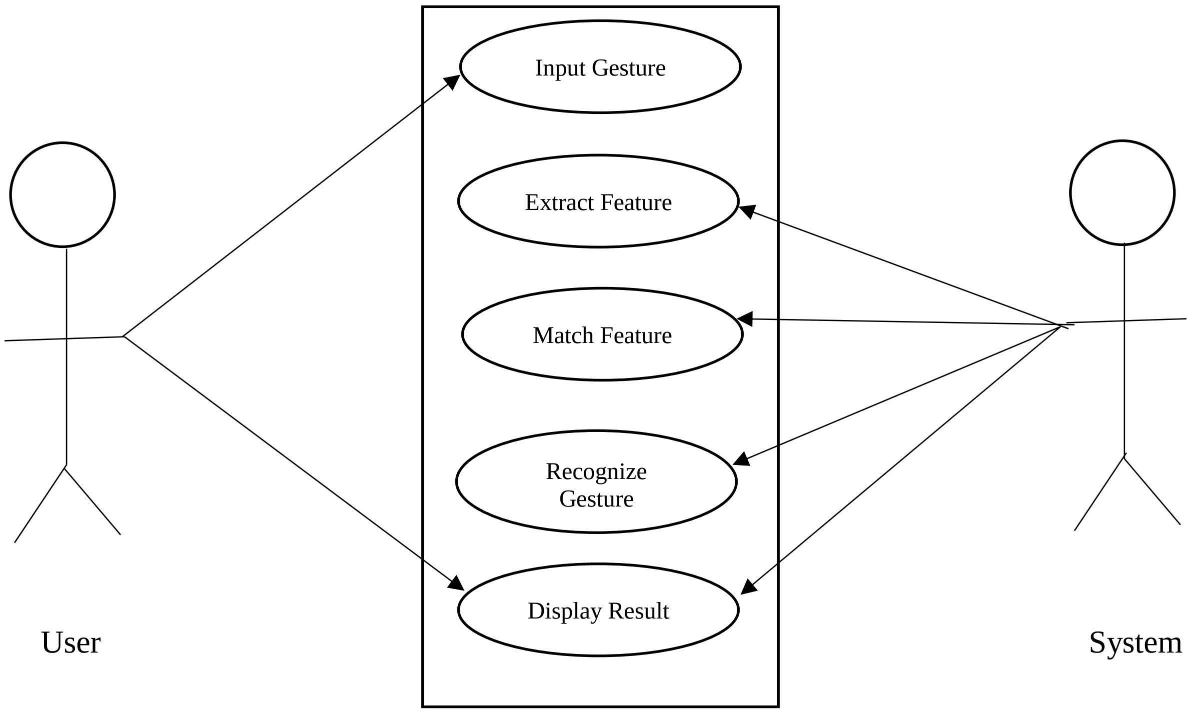

Figure 2.1 Use Case Diagram of the Hand Gesture Recognition System

2.2.1.2. Other Requirement

The user would be able to input the image and after matching process user can view the recognized gesture of hand.

2.2.2. Non-Functional Requirements

The non-functional requirements in content to the project are as follows:

2.2.2.1. Interface

Since the project is desktop-based, a desktop-based interface is required. The interface should be user friendly and easy to access by any kind of user.

2.2.2.2. Performance

The system which is developed must be able to provide information to the user and the information displayed must be accurate and correct.

2.2.2.3. Scalability

The system must provide many features (i.e. as many as possible) to provide most accurate and precise gesture recognition.

2.2.2.4. Fault Tolerance

The system must be able to cope with bad python code and errors in compiling. The system should display necessary information in case of failure.

2.2.2.5. Interoperability

In context to the project, the system should be accessible from other host and should be able to interact with other applications using standard technologies.

2.2.2.6. Usability

The applications to be developed are built in a user-friendly interface such that even a common person can understand it and make the most use of it

2.2.2.7. Maintainability

The system needs to be maintained. The recognition procedure in this project's system needs to be carried out to increase the precision of the system.

2.3. Feasibility Analysis

Feasibility studies aim to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats as presented by the environment, the resources required to carry through, and ultimately the prospects for success. The feasibility study is necessary to determine if creating a new or improved system is friendly with the cost, benefits, operation, technology and time. Following feasibility study is given as below:

2.3.1. Economic Analysis

The purpose of economic feasibility is to determine the positive economic benefits that include quantification and identification. Economic analysis is the study that determines whether a system is economically acceptable or not. In context to the project work, the system developed is a desktop application; which requires all the hardware and software support as required by other application. To build this project, economic investment will include hardware, software and manpower that have the required developing skills. According to the requirement of the project, all the resources are available. The required hardware parts and software are:

- OpenCV
- Python
- PyCharm
- Camera

2.3.2. Operational Analysis

It is concerned with the operating capabilities of the system. Since it is a desktop-based application, it is quite easy to handle the system with a normal UI skill. For the efficient operation, only a general-purpose computer is required. And the user interface is user friendly. Hence, the system is feasible operationally.

2.3.3. Technical Analysis

Technical analysis is concerned with determining how feasible a system is from a technical perspective. The project is developed for general use. In order to access the application developed in this project, the user requires a python installed. The main requirement of the system from a developer’s view are python and Camera for data. The required technologies (OpenCV of python language and any python IDE) existed. These requirements can be fulfilled. The following points were considered for the projects technical feasibility.

- The system will detect gesture from real time video.
- After detecting the gesture, it will process and give accurate class result.
- The system can detect gesture from class one to five.
- If any error is occurred system will handle it with ease.

2.4. Software and hardware Requirement

2.4.1. Software Requirement

Following are the software requirement necessary for the project.

- Python Programming Language.
- OpenCV
- Python IDE
- Windows/UNIX OS (Operating System)

2.4.2. Hardware Requirement

Following are the hardware requirement necessary for this project.

- Camera
- Computer Device (Desktop/Laptop)

2.5. Structuring System Requirements

2.5.1 Data Modeling

2.5.1.1 Classification Technique

The convexity defects for the hand contour are first calculated by using the OpenCV inbuilt function "cv.ConvexityDefects()". After the convexity defects are obtained, we perform steps for identifying the fingertips and the fingers. Convexity defects obtained, is a structure that returns four values, start point, end point, farthest point and approximate distance to farthest point, out of which three have been used.

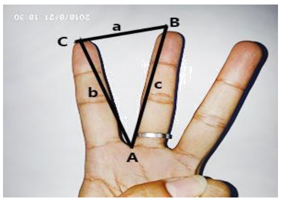

Figure 2.2 Finding Convexity Defect

The Figure 2 denotes for one of the contours the start, the end and the far point. C represents the start point, B represents the end point and A is the farthest point. The angle made by the two fingers must be found to correctly determine if a finger is held up. This is done using the triangle formed by A, B and C. The length of each line is found using the distance formula as

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

Once the length of each have been found, using the Cosine rule,

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

the angle A is found using

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

If the angle A is less than 90°, it is considered that two fingers are held up.

2.5.2. Process Modeling

2.5.2.1 Context Diagram

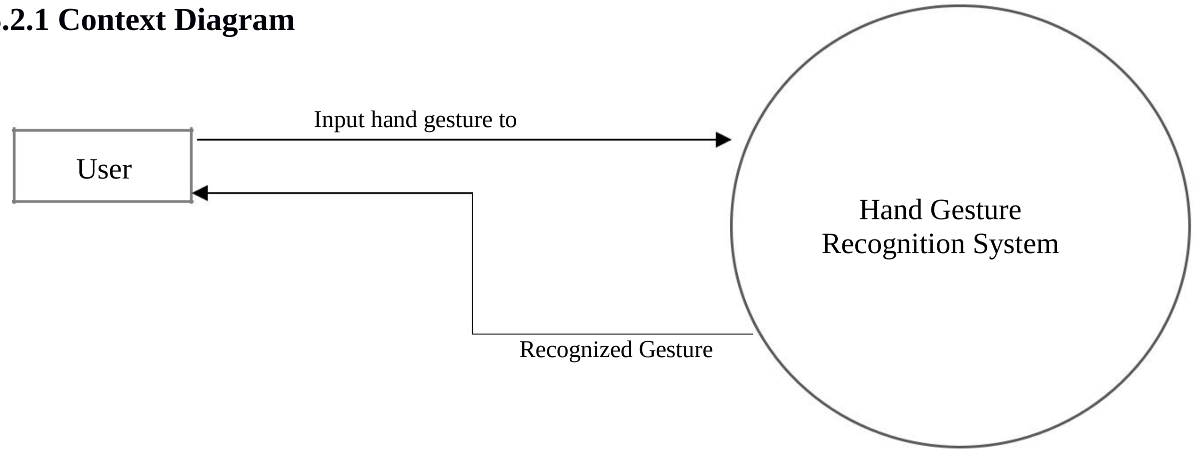

Figure 2.3 Context diagram of hand gesture recognition system

CHAPTER THREE

SYSTEM DESIGN

3.1. System Architecture

The input to the system is given using the web camera. Then the image frame will be taken in account by the system. These inputs will be preprocessed to generate appropriate condition to extract the feature from the input. The masking is done to remove noise from the frame to create environment for the high accuracy. After the preprocessing the calculation of convex hull space and contour space is calculated. Then those calculation helps to find the number of defects in the contour space. Finally, the gesture made by user is recognized and the result is displayed.

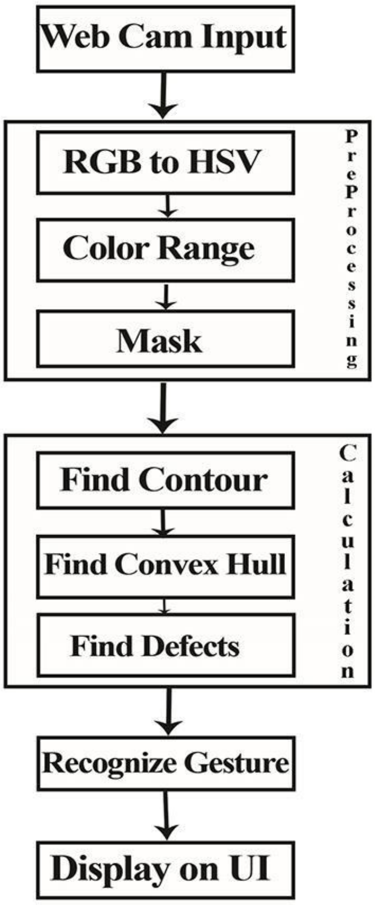

Figure 3.1 Architecture of Hand Gesture Recognition System.

3.2. Phases in Hand Gesture Recognition

The hand gesture system takes the gesture from video input. The system includes image acquisition, gesture detection, image pre-processing, calculate Contour and Convex Hull, tightness. Hand detection and feature extraction are carried out from gesture image input and then classified into five classes belonging to five basic gestures which are outlined below.

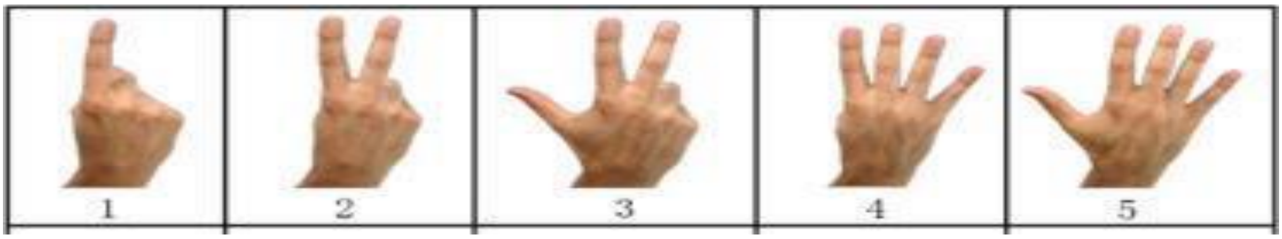

Figure 3.2 Five class of gesture from one to five

3.2.1. Image Acquisition

It is the process of receiving image for the process of recognition from the user to recognize. The image will be color, dynamic and as much as without the noise.

3.2.2. Image Preprocessing

Image preprocessing is the preliminary step of the recognition procedure. Preprocessing describes how images are normalized before extracting features from it. The region of interest is selected from given character image and make it fine conditioned. This section provides the detail description of steps used in preprocessing of given character images. The main steps of image preprocessing are given in algorithm.

Algorithm of Image Preprocessing

1. Read gesture from Webcam.
2. Convert RGB to HSV.
3. Mask the input gesture.
4. Image Inversion (making background black & foreground white).

3.2.3. Feature Extraction

In this phase, the hand gestures captured from video camera is processed in order to extract the descriptive information about it. The aim of feature extraction phase is to find and extract features that can be used to determine the meaning of a given gesture. Ideally such a feature, or a set of such features, should uniquely describe the gesture in order to achieve a reliable recognition. Some features like masking and HSV.

Example:

Input Gesture HSV Form Masked Form

Figure 3.3 Example of feature extraction

3.2.4. Contours and Convex Hull

3.2.4.1. Contours

The contour of hand is a series of points which are the boundary pixels of the hand area. After obtaining the contour, the gesture and its shape then can be detected and recognized by using contours analysis. Contour is constituted by the connection of edges, common edge recognition algorithms contain Sobel, Canny, Prewitt, Roberts, and Fuzzy logic methods. The red curve and the dots in figure 3.4. is the contour of hand gesture contained.

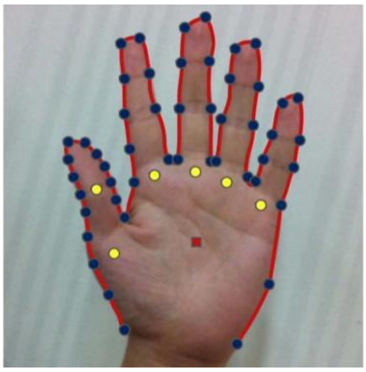

Figure 3.4 Contour

3.2.4.2. Convex Hull

The convex hull of a set of points in the Euclidean space is the smallest convex set that contains all the set of given points. For example, when this set of points is a bounded subset of the plane, the convex hull can be visualized as the shape formed by a rubber band stretched around this set of points. Convex hull is drawn around the contour of the hand, such that all contour points are within the convex hull. This makes an envelope around the hand contour.

A defect is present wherever the contour of the object is away from the convex hull drawn around the same contour. Convexity defect gives the set of values for every defect in the form of vector. This vector contains the start and end point of the line of defect in the convex hull. These points indicate indices of the coordinate points of the contour. These points can be easily retrieved by using start and end indices of the defect formed from the contour vector. Convexity defect also includes index of the depth point in the contour and its depth value from the line.

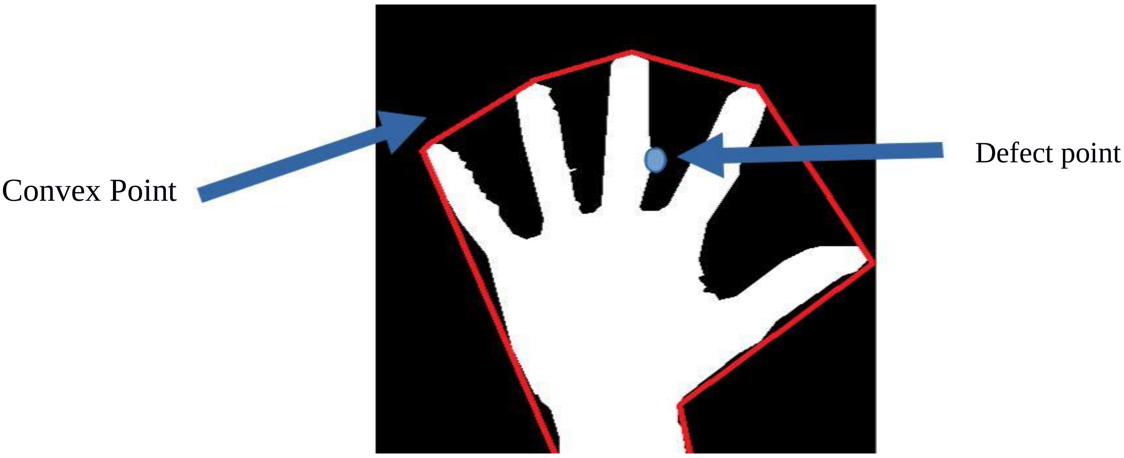

Figure 3.5 Convex hull

3.3. Process Design

The process design is as below:

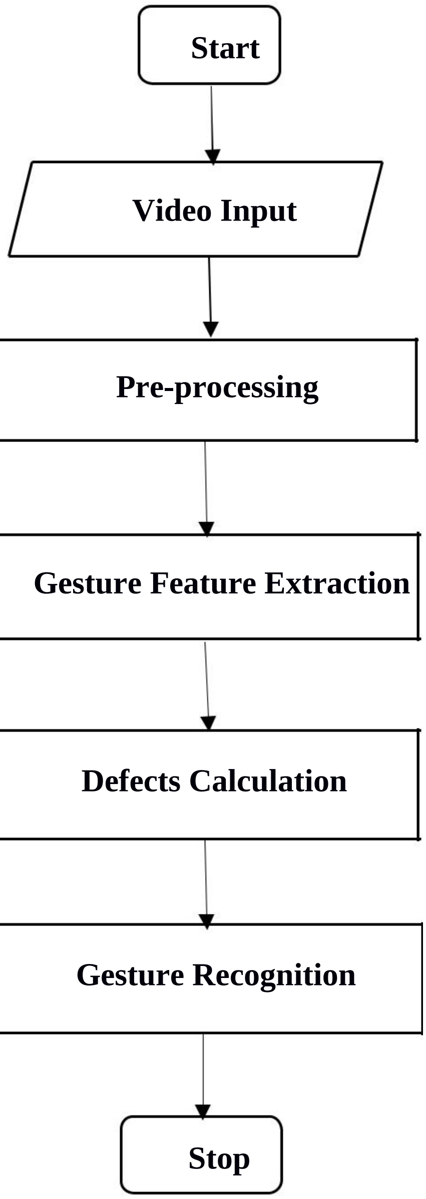

Figure 3.6 Flowchart of the system

3.4. Interface Design

We have used frame of OpenCV for the User Interface design. This frame offers customization of the look and feel of every component in an application without making significant changes to the application code. It also includes a pluggable look and feel feature, which allows it to emulate the appearance of native components while still having the advantage of platform independence. This particular feature makes writing applications in frame easy and distinguishes it from other native programs.

In interface we display the input given in the form of video, which in equivalent time reads the input as well as shows the result of input gesture. As well as we have secondary frame to display the masking process.

The UI will be like this:

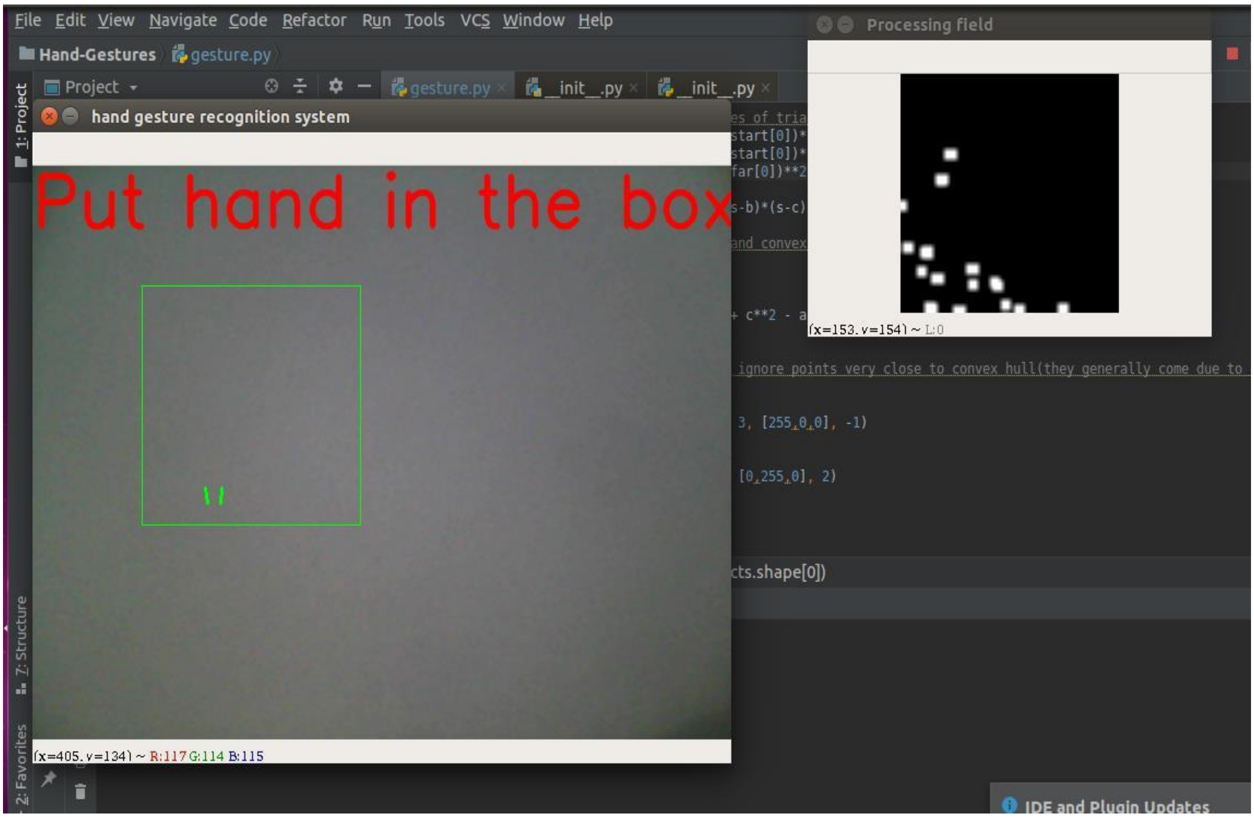

Figure 3.7 User interface of System

3.5. Input Output Design

Data in the system will be given through the video input. As we have discussed above that, image will be use to extract feature for gesture recognition. The image will be converted to the HSV then masking the image frame of respective pixel value.

There may be errors on the data input if the given input has noisy background, low quality camera and light. The gesture made by hand sometime may be out of frame so, we need to consider this type of problem too. To control these types of errors and problems we have strict input rule for the gesture input.

The output of the system will be displayed in the same UI frame/window using label.

We can see the data control as below:

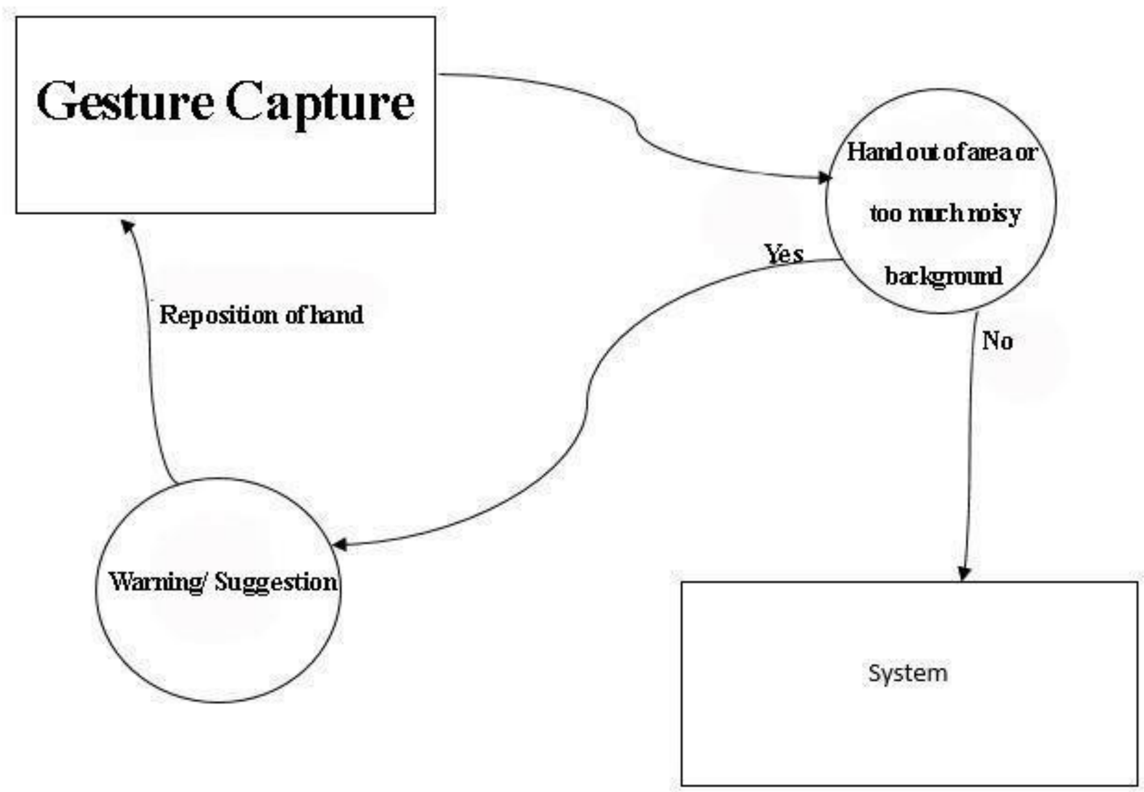

Figure 3.8 Data Control

3.6. General Information on Calculation Process

The calculation of gesture is done by analyzing features of hand. For this we find the finger tips. Firstly, we compute the convex hull for set of points corresponding to contour of hands. The convex hull is used to find the envelope of set of points of contours we detected earlier in region extraction step. In each of these convex hulls we compute the convexity defect. At each pixel j in a convex hull i, we compute the curvature of defect using vector dot product. Say [Vi(j), Vi(k)] and [Vi(l), Vi(k)] are two vectors with Vi(j) and Vi(l) representing the starting and ending point of defect, and Vi(k) is the point the deepest point of the convexity defect. Using the dot product and measuring the depth of deepest point we determine the fingertips of hand. The tightness of hand gesture contour to its convex hull is defined as the gesture convexity, which is denoted by δ, its value is the area ratio of gesture contour and convex hull.

$$\delta = (contourArea)/(\hbar ullArea)$$

Where, hullArea is the area of convex hull, contourArea is the area of gesture contour, we can get according to the image that hullArea > contourArea, so $\delta \in (0,1)$, and the bigger the value, the tighter the gesture contour to convex hull. In addition, different gestures can be distinguished by the relative position of fingertips, which is composed of two values of α and β, α is the summation of angles with centroid as vertex, lines from centroid to the first fingertip and other fingertips as edges, and β is the value of the angle with centroid as vertex, lines from centroid to the first fingertip and last fingertip as edges, that is, = 1 + 2+ ⋯ + −1 , = −1 , where N is the number of fingertips and θ is the angle between the first fingertip and the other fingertip to the gesture centroid been considered as vertex, as shown in figure 6, it has = 3, = 1 + 2 , = 2 . Figure 7 shows that different gestures have significantly different relative position of fingertips, so α, β can be used to the further recognition of different hand gestures.

After computing the features and the determining the finger tips, we are able determine whether the hand is opened or closed. An open hand satisfies three constraints they are: no of points forming a contour, depth of defect in convex hull and the other is the angle between the fingers. The number of points that forms the contour must be greater than 90 (Ni > 90).

3.7. Dialogue Design

In the process of dialogue design

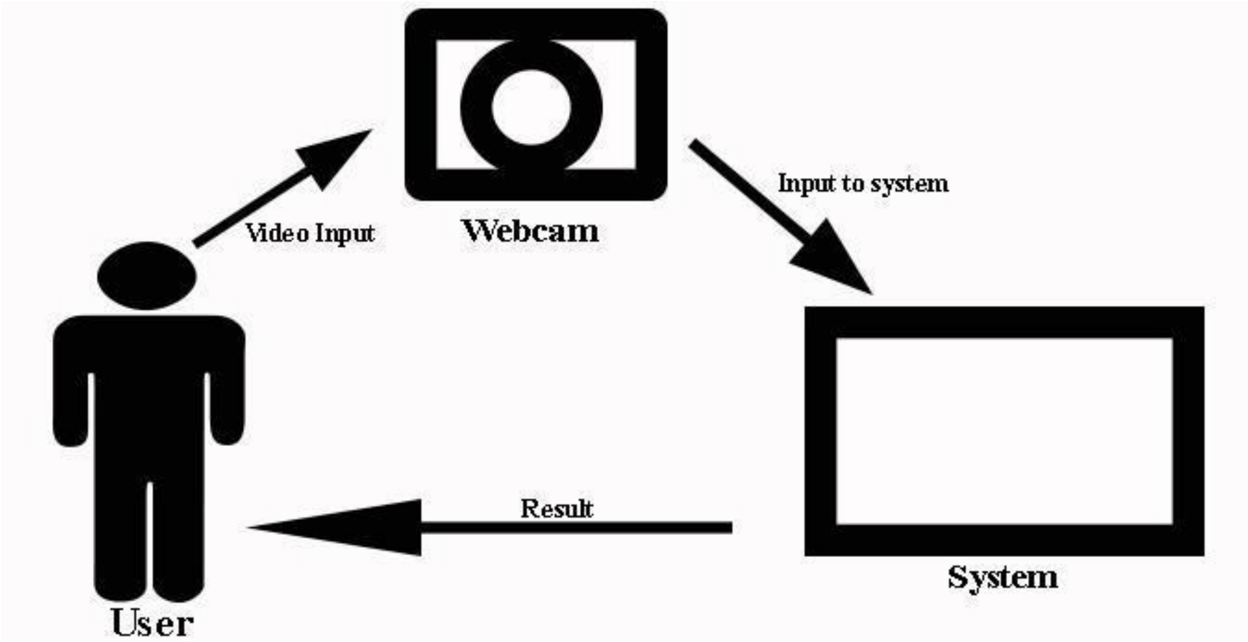

Figure 3.9 General process architecture

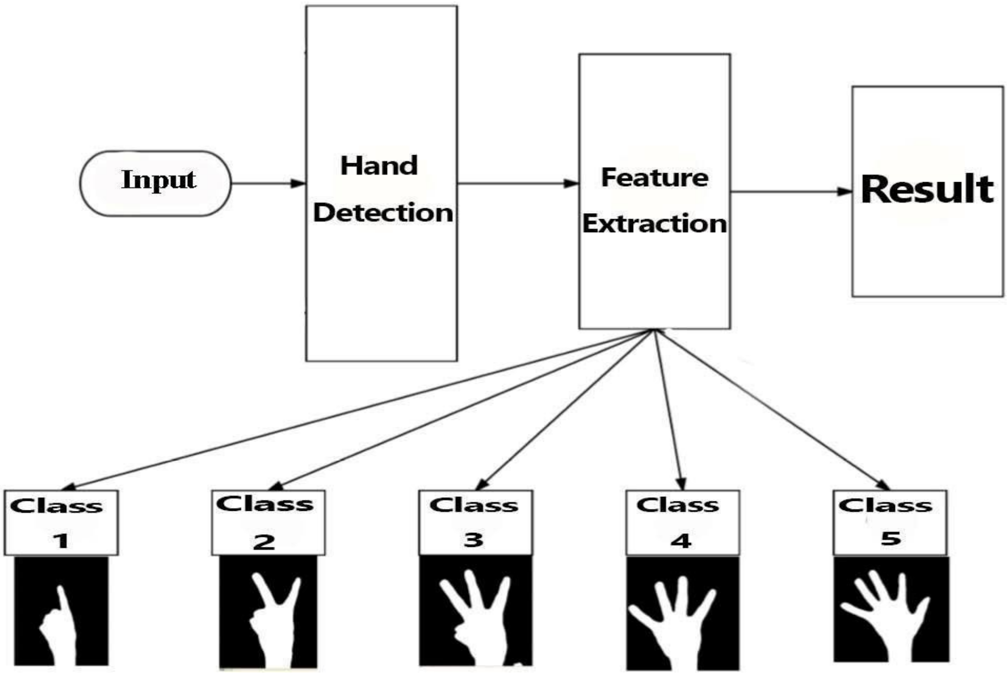

Figure 3.10 Units in the system

CHAPTER FOUR

IMPLEMENTATION AND TESTING

4.1. Implementation

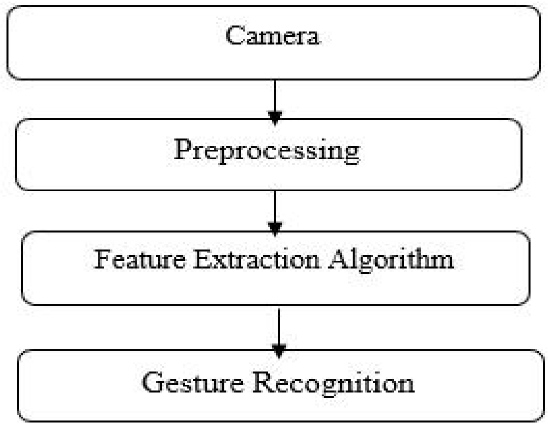

Figure 4.1 Implementation model for hand gesture system

First of all, image is taken from web-camera. Then the images are converted to HSV channel and then masked. After completing this we extract the features by calculating contour space and Hull space. Then the defects number are calculated. Finally, gesture is recognized from these particular features.

4.1.1. Tools Used

4.1.1.1. Python

Python is an interpreted high-level computer programming language for general purpose that is concurrent, class based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Python code can run on all platforms that support Python without the need for recompilation. Python applications are typically compiled to byte code. As we can get many libraries related to our project in python. We have skilled people who can guide us are highly based on the python programming field.

4.1.1.2. PyCharm-Python IDE

PyCharm is a Python integrated development environment for developing computer software. It is developed by JetBrains, and is available as an Apache 2 Licensed community edition, and in a proprietary commercial edition. It is a platform for building smart, language-aware IDEs with a comprehensive set of components, including virtual file system, text editor, UI framework, lexing, parsing, version control integration, debugger framework etc. We have used this IDE because of its ease to use. The available UI for this IDE is more user friendly. The main reason for the use of this IDE is the intelligence of the IDE itself.

4.1.1.3. OpenCV

OpenCV is a library of programming functions mainly aimed at real-time computer vision and is free for both academic and commercial use. It has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform. OpenCV supports the deep learning frameworks. We have also used OpenCV in our project for the image processing.

4.2. Testing

The testing phase determines the possible flaws and the potential inefficiency of the system.

4.2.1. Unit Testing

Unit testing refers to the testing of every small modular components of the system, keeping them isolated from other modules. Here we mention testing result of the various part of the system. In unit testing, we design the whole system in modularized pattern and each module was tested. Till we get the accurate output from the individual module we have worked on the same module. We have checked for the outcome of each module.

Table 4.1. Unit Testing

S.N	Input	Expected Result	Observed Result	Final Result
1.		1	1	1
2.		3	3	3
3.		5	5	5

4.2.2. System Testing

In this phase of testing, tests on the system passed from integrated testing are made so as to compute the hardware compatibility. Also, the tests are made more detailed to track every minute's error in the system. The primary objective of system testing is to detect if any anomalies still prevails in the system.

4.3. Accuracy Measurement

Accuracy helps to give the measurement of system's correctness. We have measure the accuracy on the basis of accurate result so within the 5 second of input gesture given. Accuracy of the system is measured as below:

Accuracy= (No. of accurate output)/ (No of input)

Table 4.2. Accuracy Measurement

No of inputs N_i	No of accurate output N_o	Accuracy=(N_o/N_i)*100%
200	197	98.5%

CHAPTER FIVE

CONCLUSION

5.1. Conclusion

The first goal of the current study was use of convex hull and counters. We mainly focused on communicating with the computer through the gestures that mainly helps to the non-vocal and deaf person to communicate with the machine. Our project study has been completely based on these techniques. We recognize the gesture based on the defect points and convex points. Our system will detect the hand gesture of five numbers.

5.2. Recommendation

The understanding of HCI and the understanding mechanisms of information processing is very needed technology for dealing with the real world. The field of HCI requires a combination of mechanical engineering, electrical engineering, and computer programming/computer science skills. The Hand Gesture Recognition System can be recommended because of:

- The system can be used to recognition the body gesture.
- The system can also be used for educational purpose.

REFRENCES

[1]. Aisha Meethian, B.M.Imran: "Real Time Gesture Recognition Using Gaussian Mixture Model", International Journal of Scientific & Engineering Research, Volume 4, Issue 8,

[2] Rajeshri R Itkarkar, Anilkumar V. Nandi "A survey of 2D and 3D imaging used in hand gesture recognition for human-computer interaction (HCI)", 15 August 2017

[3] X. Deyou, "A Network Approach for Hand Gesture Recognition in Virtual Reality Driving Training System of SPG", International Conference ICPR, pp. 519-522, 2006.

[4] E. Holden, R. Owens and G. Roy, "Hand Movement Classification Using Adaptive Fuzzy Expert System", Journal of Expert Systems, Vol. 9(4), pp. 465-480, 1996.

[5] Yanan Xu , Dong-Won Park and GouChol Pok, "Hand Gesture Recognition Based on Convex Defect Detection", International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 18 (2017) pp. 7075-7079.

[6] M. Elmezain, A. Al-Hamadi and B. Michaelis, "A Novel System for Automatic Hand Gesture Spotting and Recognition in Stereo Color Image Sequences". Journal of WSCG, Vol.17, No. 1, pp. 89-96, 2009.

[7] M. Elmezain, A. Al-Hamadi and B. Michaelis, "SpatioTemporal Feature Extraction-Based Hand Gesture Recognition for Isolated American Sign Language and Arabic Numbers".

[8]. P.Garg, N. Aggarwal, and S.Sofat,"Vision Based Hand Gesture Recognition" World Academy of Science, Engineering and Technology49, 2009.

[9]. S. khudayer ,"Hand Gesture Recognition Algorithm", Ph.D thesis, University of Technology, 2009.

[10]. e. a. Quigley, Morgan, ROS: an open-source Robot Operating System. " ICRA workshop on open source software, Vol. 3. No. 3.2.", 2009.

[11]. Asanterabi Malima, Erol Özgür, and Müjdat Çeti "A FAST ALGORITHM FOR VISION-BASED HAND GESTURE RECOGNITION FOR ROBOT CONTROL", Sabancı University, Tuzla, İstanbul, Turkey

[12]. Aseema Sultana, T Rajapuspha , "A vision based Hand gesture recognition" in the proceeding of IJCSET(2012), Vol 3,No 7

[13]. S Nagarajan, T S Subashini, “Static Hand Gesture Recognition for Sign Language Alphabets using Edge Oriented Histogram” in the proceeding of IJCA(2013), Vol 82, No 4.

[14]. Vivek D Lad, Ramesh M. Kagalkar ” Multiclass SVM Based Real-Time Hand Gesture Recognition” in the International Journal of Advanced Research in Computer and Communication Engineering

[15]. S. Mohamed Mansoor Roomi, R. Jyothi Priya and H. Jayalakshmi “Hand Gesture Recognition for Human-Computer Interaction”, in the proceedings of Journal Of Computer Science (2009).

Appendix A: Codes

```
#print the hand gestures which are in their
    ranges font =
    cv2.FONT_HERSHEY_SIMPLEX if
    l==1:
      if areacnt<2000:
        x=cv2.putText(frame,'Put hand inside box',(0,50), font, 2, (0,0,255), 3,
cv2.LINE_AA)
      else:
        if arearatio<12:
          x =cv2.putText(frame,'',(0,50), font, 2, (0,0,255), 3, cv2.LINE_AA)
        elif arearatio<17.5:
          x = cv2.putText(frame,'',(0,50), font, 2, (0,0,255), 3, cv2.LINE_AA)
       else:
          x =cv2.putText(frame,'1',(0,50), font, 2, (0,0,255), 3, cv2.LINE_AA)
elif l==2:
      x =cv2.putText(frame,'2',(0,50), font, 2, (0,0,255), 3, cv2.LINE_AA)
elif l==3:
if arearatio<27:
         x = cv2.putText(frame,'3',(0,50), font, 2, (0,0,255), 3, cv2.LINE_AA)
       else:
         x =cv2.putText(frame,'3',(0,50), font, 2, (0,0,255), 3, cv2.LINE_AA)
elif l==4:
      x = cv2.putText(frame,'4',(0,50), font, 2, (0,0,255), 3, cv2.LINE_AA)
elif l==5:
      x = cv2.putText(frame,'5',(0,50), font, 2, (0,0,255), 3, cv2.LINE_AA)
elif l==6:
      x =cv2.putText(frame,'reposition',(0,50), font, 2, (0,0,255), 3, cv2.LINE_AA)
else :
      x =cv2.putText(frame,'reposition',(10,50), font, 2, (0,0,255), 3,
cv2.LINE_AA) #return x
    #show the windows
cv2.imshow('Processing Frame',mask)
```

```
cv2.imshow('Hand Gesture Recognition System',frame)

except:
pass
        k = cv2.waitKey(5) & 0xFF
        if k == 27:
    break
      #Section for finding no. of defects
          for i in range(defects.shape[0]):
            s,e,f,d = defects[i,0]
            start = tuple(approx[s][0])
            end = tuple(approx[e][0])
            far = tuple(approx[f][0])
            pt= (100,180)

      # find length of all sides of triangle
            a = math.sqrt((end[0] - start[0])**2 + (end[1] - start[1])**2)
            b = math.sqrt((far[0] - start[0])**2 + (far[1] - start[1])**2)
            c = math.sqrt((end[0] - far[0])**2 + (end[1] - far[1])**2)
            s = (a+b+c)/2
      ar = math.sqrt(s*(s-a)*(s-b)*(s-c))
```

Appendix B: Demonstration