TITLE OF PBL PROJECT

HAND GESTURE RECOGNITON

A PBL Project Report

Submitted by:

Parangat Narsingh Pradhan (2018009122)

Rijesh Shrestha (2018011897)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHONOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

At

SHARDA UNIVERSITY

Knowledge Park III, Greater Noida, Uttar Pradesh 201310

Table of Contents

DECLARTION OF STUDENT

We Parangat Narsingh Pradhan and Rijesh Shrestha solemnly declare that the project report is based on our own work carried out during the course of our study under the supervision of Ms. Kanika Singla (SET Assistant Professor).

We assert the statements made and conclusions drawn are an outcome of our research work. We further certify that

I. The work contained in the report is original and has been done by us under the general supervision of our supervisor.

II. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad.

III. We have followed the guidelines provided by the university in writing the report.

IV. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the reference.

Parangat Narsingh Pradhan (2018009122)

Rijesh Shrestha (2018011897)

CERTIFICATE OF GUIDE

This is to certify that this project report entitled “Hand Gesture Recognition” submitted to Sharda University, is a Bonafide record of work done by Parangat Narsingh Pradhan and Rijesh Shrestha under my supervision from the start of the project to the end.

Signature of the Supervisor

Kanika Singla

Assistant Professor

Sharda University

ABSTRACT

Human-Computer interaction has developed rapidly over the past few decades. The current and the traditional way of using mouse, keyboard, joystick, etc. Has greatly supported the interaction between humans and computers but their efficiency is not very high and all of the people like differently abled people can't use them. A gesture is a movement of the hand, arms, or other body part that is intended to indicate or emphasize something. In this project we focus on studying new means of Human-Computer interaction. Our setup consists of a single camera to capture the gesture formed by the user and takes the hand input image as an input to the proposed algorithm. Due to the effect of lighting and complex background, most visual hand gesture recognition systems work only under restricted environment.

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our Faculty Guide Ms. Kanika Singla who gave us the golden opportunity to showcase our skill by this wonderful project on the topic Hand Gesture Recognition, which also helped us in doing a lot of Research and we came to know about so many new things, we are really thankful to her.

LIST OF FIGURES

1. INTRODUCTION

1.1 PROBLEM DEFINTION:

There has been vigorous research on the news modes of interaction between humans and computers over the past few decades. One of the new and most promising mode of interaction is using a camera and using hand gestures as an input. Like everything has pros and cons this method also has a few cons. The hand input must be from a well-lit background and the hand gestures also must not contain any complex background otherwise the system will not be able to read and segment the hand structure. There are also high similarities between different gestures and high differences between similar gestures. Some ways of overcoming these problems are using a high-quality camera and Kinect sensor. The other is training the system to better understand the hand input in a low lit and complex environment.

1.2 PROJECT OVERVIEW:

- Our setup consists of a single webcam from where the system will receive the input. The output of the webcam will be displayed on the monitor. The user may require to wear a white ribbon depending upon the background and how lit it is to sperate the hand from background.

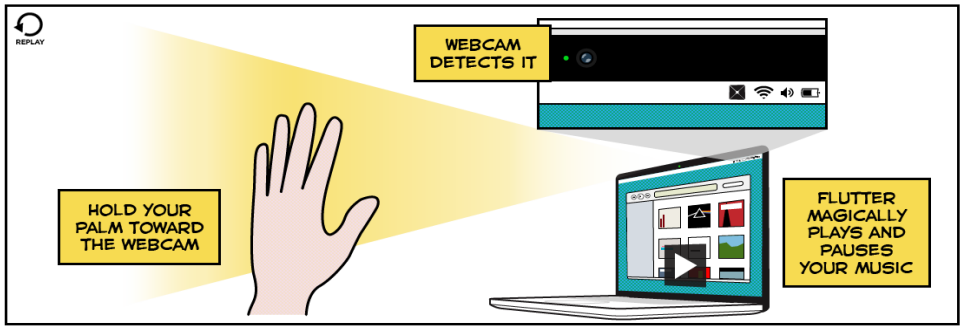

Fig 1: Basic setup for webcam-based hand gesture recognition

- After the system has captured the image of the hand. The fingers are segmented from the hand and the input is converted to a language which the system understands.

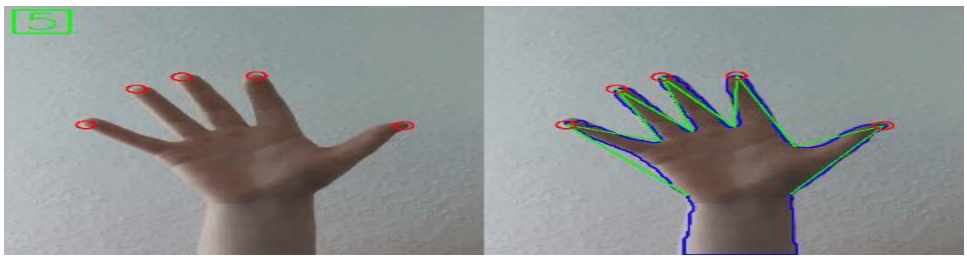

Fig 2: Fingers segmented and taken as an input

- ❖ Then the system goes through its gesture database and performs the required action.

1.3 HARDWARE REQUIREMENTS:

- All the physical equipment's i.e., input devices, processor, and output device & inter connecting processor of the computer s called as hardware.
- Hard Disk minimum of 40 GB.
- RAM minimum of 2 GB.
- Dual Core and up ,15" Monitor.
- Integrated webcam or external webcam (15 -20 fps).

1.4 SOFTWARE REQUIREMENTS:

- Operating system- Microsoft Windows 7 SP 1 or above
- Microsoft Visual Studio 2010
- MinGW and Visual C++ compilers (for Windows)
- Supporting Webcam Drivers

2. LITERATURE SURVEY

2.1 EXISTING SYSTEM:

The current way of recognizing the hand gestures by system are done manually. This results in the process being slower and the system also takes more time to give the corresponding output of the input as it is currently being trained.

2.1.1 CONS OF EXISTING SYSTEM:

- Very time consuming.
- Highly error prone
- Lot of paper work results in lot of confusion.
- Updating and Retrieval tasks are very tedious.

To avoid all these limitations and make the system working more accurately it needs to be Computerized.

2.2 PROPOSED SYSTEM:

In this section, our proposed method for recognition of hand gestures is discussed in details. The proposed system is summarized in Fig. 3. In the following, different steps of the method are explained.

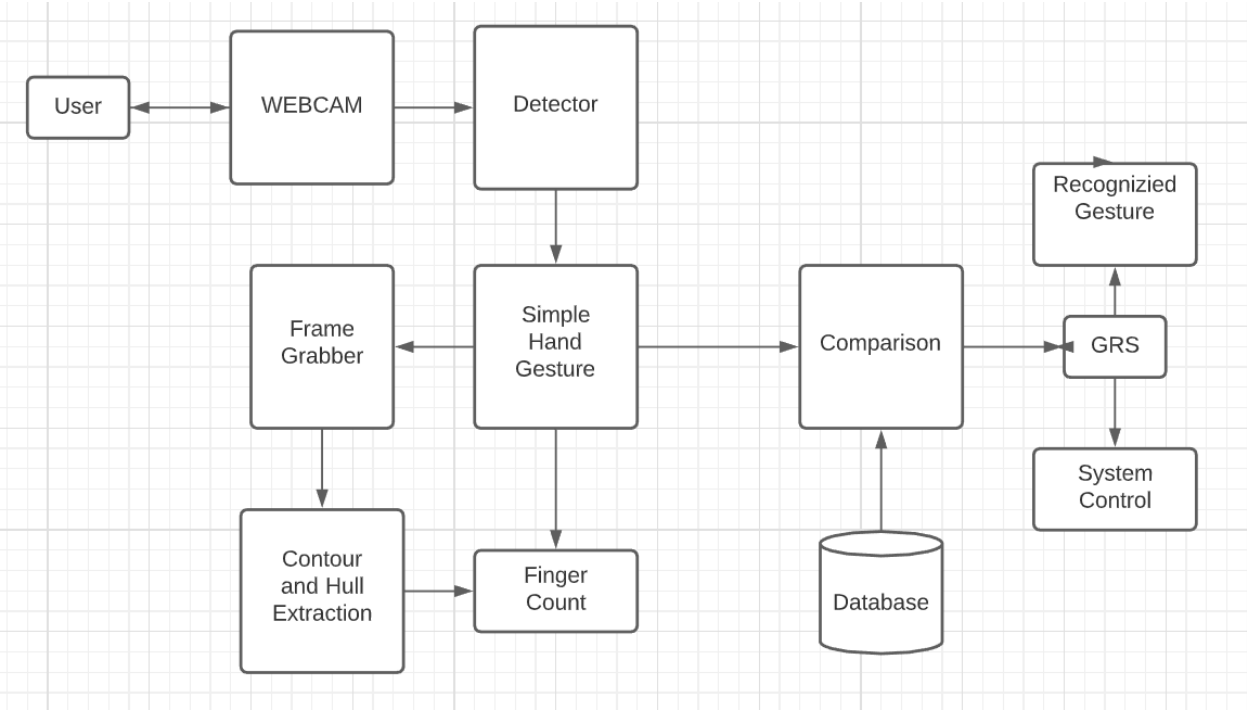

Fig. 3. Block diagram of our proposed training method

2.2.1 PROS OF PROPOSED SYSTEMS:

- Greater efficiency.
- User friendly and interactive.
- Saves lot of time.
- Ensure data accuracy

2.3 FEASIBILITY STUDY:

Human computer interaction is very important in our day-to-day life. It has become one of the basic needs of the people. Even though it is one of the basic needs for people very few advancements have been made in this field over the past few years. Keyboards, mouse, joystick, etc. is very efficient but not as efficient as we need it to be and plus some differently abled people won't be able to use it with this mode of interaction. This project aims in improving the mode of interaction by using hand gestures. Hand gestures can be taken as input to the system

through two ways either by using a Kinect sensor using HD camera or by using a single webcam and a ribbon. For our project we went with webcam to also show that HCI does not need any fancy and expensive equipment. The major drawback of using hand gesture as a mode of interaction is, we have to train the system to better understand the hand gesture as an input from a complex background and giving the desired result. We have collected some data sets through our webcam and saved it in the database and the system scans the database whenever there is an input received. Since a webcam and a ribbon is only required for this project and hours of training the system to better understand the input our project has seen some major improvements over the past few weeks. There has been vigorous testing of hand gesture recognition using expensive equipment and have seen some phenomenal improvements but these can’t be afforded by everyone so economically speaking our project is a simple yet powerful way for making Human Computer Interaction available to everyone.

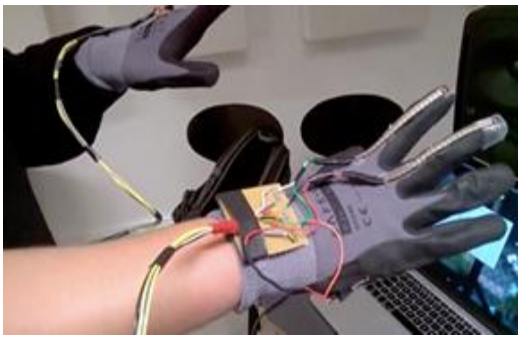

fig 4: Hand gesture recognition Using Glove consisting of Arduino is expensive

fig 5: A simple hand gesture recognition setup using only webcam

3. SYSTEM ANALYSIS & DESIGN

3.1 REQUIREMENT SPECIFICATION:

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

3.1.1 HAND DETECTION:

This software shall utilize a Hand detection system to filter out hands from the background. By applying hand detection, the system can disregard the region where the hand is located and thus reducing the amount of calculation needed to perform hand detection. The hand detection unit will be implemented with the help of OpenCV.

3.1.2 FILTERED OBJECT DETECTION:

Once the program has filtered out most of the unwanted parts of the picture after using the hand detection module, the software shall read and segment the fingers from the hand.

3.1.3 OBJECT LOCATION:

Upon detection, the system shall be able to compute the location of the object using simple trigonometry math.

3.2: FLOWCHART AND DFD:

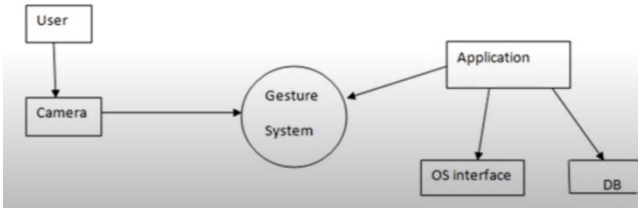

Fig 6: DFD showing how the project works

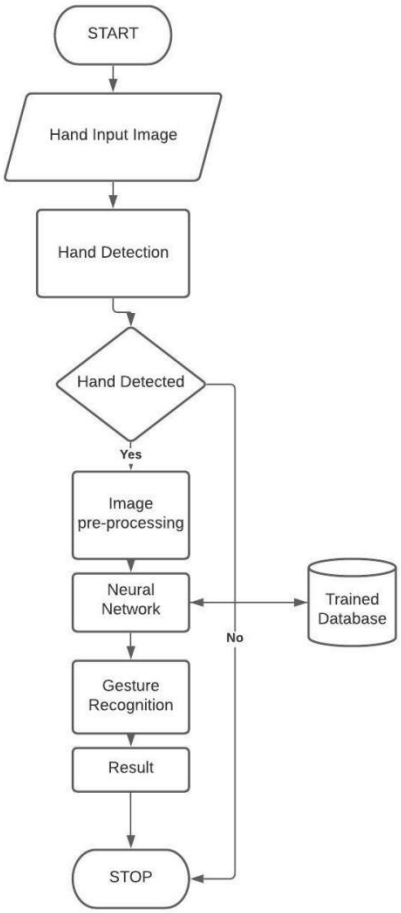

Fig 7: flowchart showing the Step by step process of of the project

3.3 STEPS TAKEN:

- Step1:

Creation of image threshold. Isolating the foreground from the Background is essential as we want the hand to be the region of Interest.

Fig 8: the threshold image of the hand is created

- Step2:

Finding Contours. The Hand is identified using an inbuilt function that finds Contours which OpenCV provides. The function is later then returning an array of co-ordinates of the formation of the Contour.

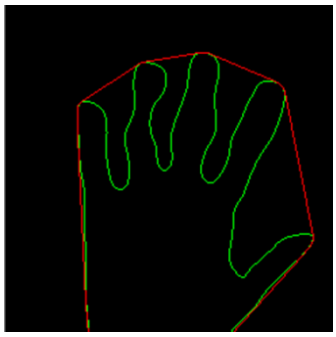

Fig 9: the fingers are segmented from the hand using Open CV

- Step3:

Convex hull and convexity defects. The data obtained from the contour's is later used to obtain entity known as "Convexity defects." They are the irregularities in contours. Based on the value of this, we can Identify how many fingers are present. This is valuable information as it provides information as to which Letter is corresponding.

Calculation: We compute a triangle. Let the sides be "a"," b" and "c". This triangle is formed by the starting point of the contour, the ending point of the contour and the farthest point of the contour. (a, b, c respectively). "a" is computed as follows

a = math.sqrt((end[0] - start[0])**2 + (end[1] - start[1])**2) [7]

Similarly, b and c are also calculated. Now using the cosine rule to calculate the angles:

$$a^2 = b^2 + c^2 - 2bc \cos A$$
$$b^2 = a^2 + c^2 - 2ac \cos B$$
$$c^2 = a^2 + b^2 - 2ab \cos C$$

$$\text{Cos A} = \frac{b^2 + c^2 - a^2}{2bc}$$
$$\text{Cos B} = \frac{a^2 + c^2 - b^2}{2ac}$$
$$\text{Cos C} = \frac{a^2 + b^2 - c^2}{2ab}$$

Fig 10: law of cosines

- Step4:

Identification.

Letter A: For identifying A, we computed the difference between the area of a circle and the area of the contour. The circle is obtained by bounding the contour. The reason this method is adopted for A is that there is very little difference between the two areas (mentioned above) which makes the Letter A stand out from the other letters. Hence this algorithm was found to be very efficient.

Letter B: For the letter B, we computed the contour area. This method is adopted because the Letter B has the largest area among the other letters.

Letters V, C, L, Y: This part gets executed when the Letter A fails. If the number of convexity defects are equal to 1, the following algorithm is employed. The "angle" is calculated. This entity is obtained by an OpenCV inbuilt function that calculates the overall figure's orientation, giving us an angle. Based on the values of the angle, Letters V, C, L and Y are identified.

Letters F and W: Letters F and W are the only alphabets in the American Sign Language to have 2 convexity defects. Once 2 convexity defects are identified, the angle is compared. Hence the Letters F and W are identified in this manner.

Letters D, J, H, I, U: A combination of parameters are computed to identify these letters, Solidity, Aspect ratio and Angles are computed. On intensive testing of Contour Parameters, we found the above parameters to be reliable and efficient.

- Step5:

Converting into Audio files. Once a corresponding Letter is identified, I executed a function I defined. The algorithm is given as follows. For example, the letter "U" is identified, I make a directory and navigate into that directory. Then a ".txt" file is made containing the "U". If U already exists, it is overwritten. Next, I wrote a Shell Script which goes into the directory and finds the last modified file, using its corresponding time stamp. Once the last modified file is found, the script plays the corresponding audio file. This approach doesn't depend on the python program.

4. REFERENCES

1. Hand Gesture Recognition System Using Camera

Viraj Shinde, Tushar Bacchav, Jitendra Pawar, Mangesh Sanap

B.E computer engineering, Navsahyadri Education Society's Group of Institutions, pune.

2. Review of Hand Gesture Recognition Study and Application

Yanan Xu, Yunhai Dai

3. Hand gesture recognition system

Prof. Praveen D. Hasalkar, Rohit S. Chougule, Vrushabh B. Madake, Vishal S. Magdum4 Department of Computer Science and Engineering, Walchand Institute of Technology.

4.OpenCV Tutorials — OpenCV 3.0.0-dev documentation.

5.Google dictionaries.

6. 3.9.0 Documentation (python.org)