

# **SIGN LANGUAGE DECODER FOR THE HEARING AND SPEAKING IMPAIRED**

Submitted in partial fulfillment of the requirements for the award of  
Bachelor of Engineering degree in Computer Science and  
Engineering by

**JOSHUA KENNETH PAUL J (Reg. No. 37110297)**

**HARIHARAN A (Reg. No. 37110266)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
SCHOOL OF COMPUTING**

## **SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC  
JEPPIAAR NAGAR, RAJIV GANDHI  
SALAI, CHENNAI – 600 119**

**MARCH - 2021**



# **SATHYABAMA**



INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

Accredited with "A" grade by NAAC  
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

[www.sathyabama.ac.in](http://www.sathyabama.ac.in)

## **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

### **BONAFIDE CERTIFICATE**

This is to certify that this project report is the bonafide work of **JOSHUA KENNETH PAUL J (Reg. No. 37110297)** and **HARIHARAN A (Reg. No.37110266)** who carried out the project entitled "**SIGN LANGUAGE DECODER FOR THE HEARING AND SPEAKING IMPAIRED**" under my supervision from **NOVEMBER 2020** to **March 2021**.

Internal Guide  
**S.L. JANY SHABU M.Tech.,(PhD)**

Head of the Department

---

**Submitted for Viva voce Examination held on**\_\_\_\_\_

**Internal Examiner**

**External Examiner**

## **DECLARATION**

I, **JOSHUA KENNETH PAUL J** hereby declare that the Project Report entitled  
**“SIGN LANGUAGE DECODER FOR THE HEARING AND SPEAKING  
IMPAIRED”**

is done by me under the guidance of **Mrs. S.L. JANY SHABU M.Tech.,(PhD)**,  
Department of Computer Science and Engineering at Sathyabama Institute of  
Science and Technology is submitted in partial fulfillment of the requirements for  
the award of Bachelor of Engineering degree in Computer Science and  
Engineering.

**DATE:**

**PLACE: CHENNAI**

**SIGNATURE OF THE CANDIDATE**

## ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mrs. R. Sathya Bama Krishna, M.E., Assistant Professor**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

## **ABSTRACT**

Sign language is the only medium of contact for deaf and dumb people. These visually disabled people communicate their feelings and ideas to other persons with the aid of sign language. As it is difficult for the normal human to understand these languages, these mentally disabled individuals always have to hold the interpreter along for them to connect with the world. Recognition of sign language has since become an analytical assignment. Since, sign language contains a variety motions and hand gestures, the consistency of sign language relies on the exact recognition of hand gestures. This paper presents a novel approach for sign language recognition. The main objective of the paper is recognition and classification of hand gesture with maximum accuracy. This paper proposes a novel approach in skin color-based segmentation for hand region detection from background, Histogram of Gradients (HOG) method which is used for extracts the shape information features and finally Random Forrest machine learning classifier is utilized to sign language recognition with effective results. The accuracy of our new models was found to be considerably higher than that of the current one. This technology will help Employers in recruiting them and the dumb and deaf person can present his thoughts in large seminars and conferences.

# TABLE OF CONTENTS

| CHAPTER<br>NO. | TITLE                                        | PAGE<br>NO. |
|----------------|----------------------------------------------|-------------|
|                | <b>ABSTRACT</b>                              |             |
|                | <b>v</b>                                     |             |
|                | <b>LIST OF FIGURES</b>                       | <b>viii</b> |
|                | <b>LIST OF TABLES</b>                        | <b>x</b>    |
|                | <b>LIST OF ABBREVIATION</b>                  | <b>xi</b>   |
| <b>1</b>       | <b>INTRODUCTION</b>                          | <b>1</b>    |
|                | 1.1 INTRODUCTION                             | 1           |
|                | 1.2 HISTORY OF SIGN LANGUAGE                 | 1           |
|                | 1.3 INDIAN SIGN LANGUAGE                     | 2           |
| <b>2</b>       | <b>LITERATURE SURVEY</b>                     | <b>4</b>    |
|                | 2.1 LITERATURE REVIEW                        | 4           |
| <b>3</b>       | <b>AIM AND SCOPE</b>                         | <b>7</b>    |
|                | 3.1 AIM                                      | 7           |
|                | 3.2 OBJECTIVES                               | 7           |
|                | 3.3 SCOPE                                    | 7           |
|                | 3.4 ADVANTAGES                               | 7           |
| <b>4</b>       | <b>METHODS AND ALGORITHMS</b>                | <b>9</b>    |
|                | 4.1 METHODOLOGY                              | 9           |
|                | 4.1.1 IMAGE PROCESSING                       | 9           |
|                | 4.1.2 SUPPORT VECTOR MACHINE ALGORITHM       | 10          |
|                | 4.1.2.1 ADVANTAGES OF SUPERVISED<br>LEARNING |             |

| CHAPTER<br>NO. | TITLE                                 | PAGE | NO.       |
|----------------|---------------------------------------|------|-----------|
|                | 4.1.3 GAUSSIAN MIXTURE MODEL          |      | 11        |
|                | 4.1.4 HISTOGRAM OF ORIENTED GRADIENTS |      | 13        |
|                | 4.1.5 RANDOM FOREST CLASSIFIER        |      | 14        |
| 4.2            | <b>PROPOSED SYSTEM</b>                |      | <b>16</b> |
| 4.3            | <b>MODULES</b>                        |      | <b>16</b> |
|                | 4.3.1 BACKGROUND SUBTRACTION          |      | 16        |
|                | 4.3.2 FEATURE EXTRACTION              |      | 18        |
|                | 4.3.3 CLASSIFICATION                  |      | 20        |
| <b>5</b>       | <b>RESULT AND DISCUSSIONS</b>         |      | <b>23</b> |
|                | 5.1 RESULTS                           |      | 23        |
| <b>6</b>       | <b>CONCLUSION AND FUTURE WORK</b>     |      | <b>28</b> |
|                | 6.1 CONCLUSION                        |      | 28        |
|                | 6.2 FUTURE WORKS                      |      | 28        |
|                | <b>REFERENCES</b>                     |      | <b>29</b> |
|                | <b>APPENDIX</b>                       |      | <b>31</b> |
|                | A SOURCE CODE                         |      | 31        |
|                | B SCREENSHOTS                         |      | 40        |
|                | C PAPER PUBLICATION                   |      | 53        |
|                | D PLAGIARISM REPORT                   |      | 60        |

## LIST OF FIGURES

| FIGURE NO. | FIGURE NAME                                | PAGE NO. |
|------------|--------------------------------------------|----------|
| 1.1        | Samples of Indian Sign language            | 2        |
| 4.1        | Steps involved in SVM                      | 10       |
| 4.2        | Working of SVM Classifier                  | 11       |
| 4.3        | Formula's in GMM                           | 12       |
| 4.4        | Working process in GMM                     | 12       |
| 4.5        | HOG recognizes Hand gesture                | 14       |
| 4.6        | Representation of Random Forest Classifier | 15       |
| 4.7        | Step 1 in Background Subtraction           | 17       |
| 4.8        | Step 2 in Background Subtraction           | 18       |
| 4.9        | Step 3 in Background Subtraction           | 18       |
| 4.10       | Working of Decision Tree                   | 22       |
| 5.1        | Dataset Sample for "Call"                  | 23       |
| 5.2        | Dataset Sample for "Go"                    | 23       |
| 5.3        | Dataset Sample for "One"                   | 23       |
| 5.4        | Dataset Sample for "Five"                  | 24       |
| 5.5        | Dataset Sample for "Looser"                | 24       |
| 5.6        | Dataset Sample for "Two"                   | 24       |
| 5.7        | Dataset Sample for "Stop"                  | 24       |
| 5.8        | Dataset Sample for "Thumbs up"             | 24       |
| 5.9        | Complete data set of call me               | 25       |
| 5.10       | Complete data set of five                  | 26       |
| 5.11       | Final output of Call me                    | 26       |

| <b>FIGURE<br/>NO.</b> | <b>FIGURE NAME</b>          | <b>PAGE<br/>NO.</b> |
|-----------------------|-----------------------------|---------------------|
| 5.12                  | Final Output of “Five”      | 27                  |
| 5.13                  | Final Output of “Stop”      | 27                  |
| 5.14                  | Final Output of “Looser”    | 28                  |
| 5.15                  | Final Output of “One”       | 28                  |
| 5.16                  | Final Output of “Stop”      | 28                  |
| 5.17                  | Final Output of “Thumbs up” | 28                  |
| 5.18                  | Final Output of “Two”       | 29                  |

## LIST OF TABLES

| TABLE NO. | TABLE NAME                        | PAGE NO. |
|-----------|-----------------------------------|----------|
| 4.1       | REPRESENTATION OF A DIGITAL IMAGE | 9        |

## LIST OF ABBREVIATIONS

| ABBREVIATION | EXPANSION                         |
|--------------|-----------------------------------|
| ARL          | ARMY RESEARCH LABORATORY          |
| BGR          | BLUE-GREEN-RED                    |
| CCTV         | CLOSED-CIRCUIT TELEVISION         |
| CNN          | CONVOLUTION NEURAL NETWORKS       |
| CR           | CONTAINER REGISTRY                |
| CSS          | CASCADING STYLE SHEETS            |
| CV           | COMPUTER VISION                   |
| EUA          | END USER APPLICATION              |
| GCP          | GOOGLE CLOUD PLATFORM             |
| HTML         | HYPertext MARKUP LANGUAGE         |
| HOG          | HISTOGRAM OF ORIENTED GRADIENTS   |
| JS           | JAVASCRIPT                        |
| JSON         | JAVASCRIPT OBJECT NOTATION        |
| PCA          | PRINCIPAL COMPONENT ANALYSIS      |
| RGB          | RED-GREEN-BLUE                    |
| SIFT         | SCALE INVARIANT FEATURE TRANSFORM |
| SURF         | SPEEDED-UP ROBUST FEATURE         |
| SVM          | SUPPORT VECTOR MACHINE            |
| VM           | VIRTUAL MACHINE                   |

# **CHAPTER 1**

## **INTRODUCTION**

### **N**

#### **1.1 INTRODUCTION**

A sign language is a way of communicating by using the hands and other parts of the body. Sign languages are an important way for deaf people to communicate. Deaf people often use them instead of spoken languages. Spoken languages use sounds from the mouth and are understood with the ears. Sign languages use hands and are understood with the eyes. Deaf people can use sign languages more easily than spoken languages. Deaf people sometimes learn a sign language from their family, especially if their parents are deaf. But most deaf children have hearing parents, so they learn a sign language from other deaf people. They may meet other deaf people at school or in the streets. Hearing people may learn to sign directly from deaf people. Or, they may learn a sign language by going to signing classes or by studying a sign language workbook, which can come with an interactive DVD, YouTube. Sometimes deaf people do use a spoken language, especially when talking with hearing people. Sometimes hearing people use a sign language with each other, rather than speaking. But deaf people tend to use sign languages, and hearing people tend to use spoken languages. Some deaf people can also understand spoken words by looking at a speaker's lips. This is known as lip-reading. It is hard to learn, and few people do it well. Sometimes signing and lip-reading are combined, especially when deaf and hearing people are talking to each other.

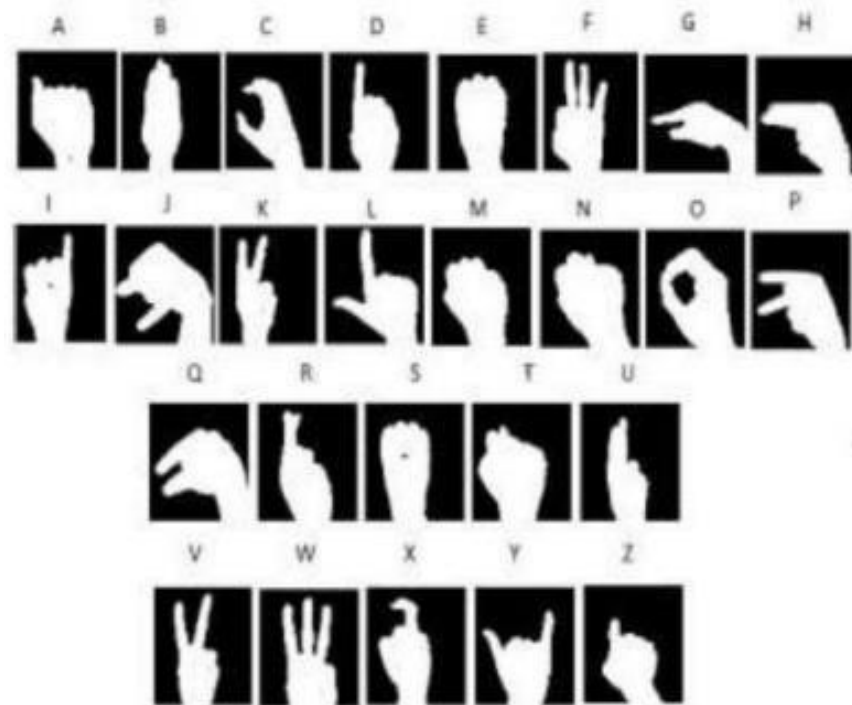
#### **1.2 History of Sign Language**

The recorded history of sign language in Western societies starts in the 17th century, as a visual language or method of communication. Sign language is composed of a system of conventional gestures, mimic, hand signs and finger spelling, plus the use of hand positions to represent the letters of the alphabet. Signs can also represent complete ideas or phrases, not only individual words. Most sign languages are natural languages, different in construction from oral languages used in proximity to them, and are employed

mainly by deaf people in order to communicate.

Until the 19th century, most of what we know about historical sign languages is limited to the manual alphabets (finger-spelling systems) that were invented to facilitate transfer of words from an oral to a sign language, rather than documentation of the sign language itself. Many sign languages have developed independently throughout the world, and no first sign language can be identified. Both signed systems and manual alphabets were found worldwide, and, though most recorded instances of sign languages seem to occur in Europe in the 17th century, it is possible that popular European ideals have overshadowed much of the attention earlier signed systems may have otherwise received. It was commonly accepted, for instance, that “the deaf” could not be educated

### 1.3 INDIAN SIGN LANGUAGE



**Fig 1.1 Samples of Indian Sign Language**

India has its own sign language, Indian Sign Language, under the name (ISL). A few schools for deaf children remain in developed nations. In developed countries, unemployment rates for adults with hearing loss are very high. According to data from Ethnologue, the literacy rate and number of children attending school are much lower among the deaf community in India, which is around 1 percent of the total population. It

goes on to note that official acceptance of sign languages dramatically boosts usability, enhances the availability of

interpreters and offers documentation of sign languages. Finger spelling is the illustration of a writing system's letters and occasionally numeral systems. Using finger spelling, Sign Language (ISL) will represent the English alphabet A-Z. It can be one handed or two handed, and the two-handed form is followed by ISL. Several scholars have suggested various forms of hand gesture edge detection techniques in the last year, but these techniques have demonstrated several disadvantages, such as false edge detection, broken edges, high calculation time, etc. In terms of edge configuration, edge orientation and noise setting, the set of edge detection variables is constrained.

## **CHAPTER – 2**

### **LITERATURE SURVEY**

#### **2.1 LITERATURE REVIEW**

1. Title: A Machine Learning Based Approach for Hand Gesture Recognition using Distinctive Feature Extraction

Author: Himadri Nath Saha, Sayan Tapadar, Shinjini Ray, Suhrid Krishna Chatterjee, Department of Electronics and Communication Engineering, Institute of Engineering and Management, Kolkata, India

Abstract: In a world of almost 7 billion people more than 500 million suffer from some physical, sensory or mental disability. Their lives are often impeded by such deformities which bars them from full participation in society and the enjoyment of equal rights and opportunities. Sign language is common for the deaf and the dumb. Sign language is an efficient alternative to talking, where the former is replaced by hand gestures. Hand gestures are combination of hand shapes, orientations and movement of the hands, alignments of the fingers and positioning of the palm which are used to express fluidly a conveyer's thoughts. Signs are used to communicate words and sentences to audience. The objective of this paper is to optimize an algorithm for recognition of hand gestures with reasonable accuracy, where the input to the pattern recognition system will be given from the hand.

2. Title: Blind/Deaf Communication API for Assisted Translated Educational Digital Content

Author: João Ulisses, Tiago Oliveira, Emanuel Rocha, Paula Maria Escudeiro, ISEP - School of Engineering, Polytechnic Institute of Porto, Port

Abstract: With the rise of usage in digital content in education, deaf and blind communities face communication barriers which as a result makes education less inclusive. These barriers do not allow them to integrate within the larger scholarly

communities as most tools used for information dissemination remain inaccessible to them. This paper presents BDC-API (Blind/Deaf Communications API), a free-to-use modular toolkit that will ease

accessibility for the blind and deaf communities to digital education content. This content includes the use cases of Massive Online Open Courses and Serious Games used in education. BDC-API incorporates the use of state of the art technologies such as, 3D sign language translator, grammar translation, voice recognition and text-to-speech. This paper demonstrates in greater detail, how these technologies culminate in the creation of an API ready to use for any educational digital content and how the BDC-API can ensure higher quality of digital content.

3. Title:\_Hand Gesture Recognition Using K-Means Clustering and Support Vector Machine

Author: DeviraAnggi Maharani,Hanif Fakhurroja, Riyanto and Carmadi Machbub,School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Bandung, Indonesia

Abstract: Human-Robot Interaction (HRI) requires media for communication which can be both understood by robot and easily done by human. Usually, human using oral language to communicate but there are some situations that require performing non-verbal activities such as deaf people, patient, and old people, therefore gesture recognition as communication media is needed to give order to Robot. This paper discusses hand gesture recognition as input command for Bioloid Premium Robot using two methods, K-Means clustering and Support Vector Machine (SVM) with directed acyclic graph (DAG)decision. Four gestures (forward, right, left and stop) were recognized using Kinect v2. The testing was done 6 peoples for three distances (2m, 3m, and 4m) and three slopes position (45, 0,-45). The SVM required 10ms recognition time with accuracy reached 95.15%, while K-Means needed 4.45ms recognition time with 77.42% accuracy. This study resulted in Multiclass SVM with DAG decision performs better than K-Means clustering method.

4. Title:\_Sign-Voice Bidirectional Communication System for Normal

Author:\_MariamMoustafa Reda, Nada Gamal Mohammed , Rania Ahmed Abdel Azeem Abul Seoud,Department of Computer Science, Faculty of Computers and Information, Fayoum University

Abstract: "deaf/dumb" and blind people face problems in communicating with others with difficulties in dealing with the communication technology. The goal of this paper is to design a desktop human computer interface application that is used to facilitate communication between normal, "deaf/dumb" and blind people. SVBi Comm system helps blind person to hear voice saying the word gestured by the "deaf/dumb" while the deaf will receive a gesture representing the word said by the blind. SVBi Comm works in two directions, the first direction is processing from video to speech. The animated word gestures are mapped with language knowledge base into text. Then, the relevant audio is generated using Text-to-Speech (TTS) API. The second direction is processing from speech to video. The voice from blind is converted into its corresponding text using Speech to-Text (STT) API. Then, the natural language is mapped from the database to "deaf/dumb" in a relevant sign language form by using a 3D graphical model. The system was evaluated using a set of 113 sentences with 244 signs. In voice recognition; system recognized words with a percentage of 90% from 19 different persons. For Image recognition the system recognized images with a percentage of 84% for 21 different persons. SVBi Comm system provides many facilities with low cost that could be used in many areas.

##### 5. Title: Moment Based Sign Language Recognition for Indian

Languages Author: Umang Patel, Aarti G. Ambekar

Abstract: Communication plays a major role in day-to-day life. But it is very difficult for normal people to communicate with deaf, dumb & blind people & vice versa. Because of a deaf, dumb use sign language for communication and normal people are unable to understand sign language. So there is a need of converting sign language. This paper minimizes the barrier of communication between them. In this paper, Hand gestures are captured, processed with the help of MATLAB and then converted into speech & text. For speech & text, two languages are chosen English & Hindi. In this paper features value of images is evaluated based on moment technique. Two classification techniques (PNN & KNN) are used & performance is compared between both classifiers. This paper will help to do the easiest communication between normal &

speech impaired people.

## **CHAPTER – 3**

### **AIM AND**

### **SCOPE**

#### **3.1 AIM OF PROJECT**

The primary aim of this proposed system is that In recent year several authors have proposed different types of hand gesture edge detection techniques, but those methods have shown several drawbacks like false edge detection, broken edges, high computation time etc. Since selection of edge detection variables is limited in term of edge structure, edge orientation, and noise environment. The proposed system provides a novel way in identifying the hand gesture with high accuracy with less computational time.

#### **3.2 OBJECTIVES**

- To recognize the hand gesture to identify the sign language.
- To increase the detection accuracy.
- To decrease the computational time for recognition.
- To create a lightweight application that doesn't require a powerful and expensive processing hardware.
- To make it user friendly so the impaired person can navigate the application without any complexity.
- To create an interface which the non-impaired individual can understand without any confusion.

#### **3.3 SCOPE**

This project will help people with speech and hearing deficiencies to be able to achieve simple communication in sign language, since it is their most comfortable language, with others for basic everyday tasks.

### **3.4 ADVANTAGES**

- Time saving as it is operated online.
- There is need of any external apparatus such as a pen and paper.
- Being an offline application, it can be used in places with no connectivity.
- No heavy processing power required, can be easily run on a mobile processor, since it's a lightweight application.
- Modern image recognition techniques and python programming language is used for development with future enhancements in mind.
- A simple interface helps prevent any complications in understanding the application.

## CHAPTER – 4

### METHOD AND ALGORITHM

#### 4.1 METHODOLOGY

##### 4.1.1 IMAGE PROCESSING

**IMAGE PROCESSING** Digital image processing deals with developing a digital system that performs operations on a digital image. An image is nothing more than a two-dimensional signal. It is defined by the mathematical function  $f(x,y)$  where  $x$  and  $y$  are the two coordinates horizontally and vertically. The value of  $f(x,y)$  at any point gives the pixel value at that point of an image. An image is nothing but a two-dimensional array of numbers ranging between 0 and 255.

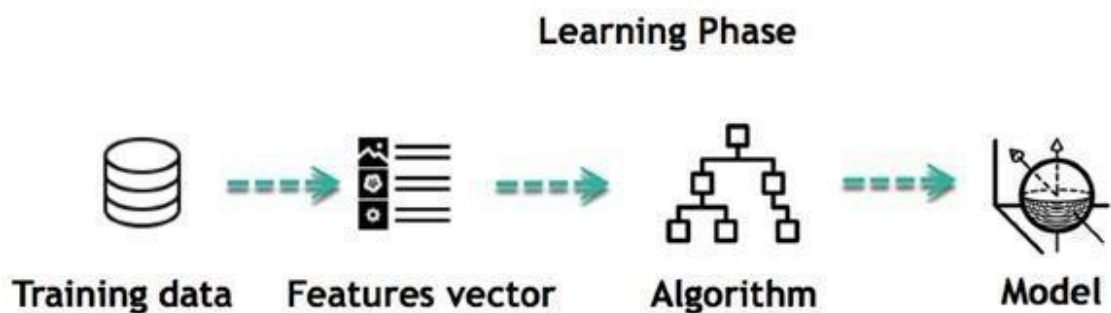
|     |     |     |
|-----|-----|-----|
| 128 | 30  | 123 |
| 232 | 123 | 321 |
| 123 | 77  | 89  |
| 80  | 255 | 255 |

***Table 4.1 Representation of a Digital Image***

Each number represents the value of the function  $f(x,y)$  at any point. In this case the value 128 , 230 ,123 each represents an individual pixel value. The dimensions of the picture is actually the dimensions of this two dimensional array. 15 In the physical world, any quantity measurable through time over space or any higher dimension can be taken as a signal. A signal is a mathematical function, and it conveys some information. The two-dimensional signals are those that are measured over some other physical quantities such as a digital image.

#### 4.1.2 SUPPORT VECTOR MACHINE ALGORITHM (SVM)

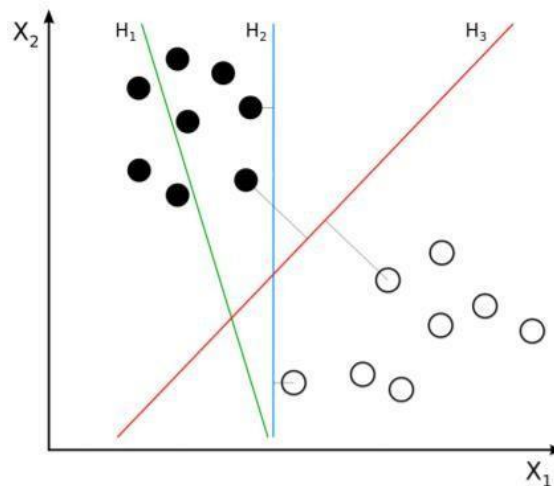
Support Vector Machine (SVM) is a supervised machine learning model used for two- group classification problems. After giving an SVM model set of labeled training data for each category, they're able to categorize new test data. SVM classifies data based on the plane that maximizes the margin. The SVM decision boundary is straight. SVM is a really good algorithm for image classification. Experimental results show that SVMs achieve significantly higher search accuracy than traditional query refinement schemes after just three to four rounds of relevant feedback. This is also true for image segmentation systems, including those using a modified version SVM that uses the privileged approach. The SVM classifier algorithm is used to compare and distinguish each person's face using the face encodings from each image by comparing the face distance (i.e., data points) between the faces to conclude the result.



**Fig 4.1 Steps involved in SVM**

**Supervised Machine Learning** The majority of practical machine learning uses supervised learning. Supervised learning is where you have input variables ( $x$ ) and an output variable ( $Y$ ) and you use an algorithm to learn the mapping function from the input to the output.  $Y = f(X)$  The goal is to approximate the mapping function so well that when you have new input data ( $x$ ) that you can predict the output variables ( $Y$ ) for that data. It is called supervised learning because the process of an algorithm learning from the training dataset can be thought of as a teacher supervising the learning process. We know the correct answers, the algorithm iteratively makes predictions on the training data and is corrected by the teacher. Learning stops when the algorithm achieves an acceptable level of performance.

Supervised learning problems can be further grouped into regression and classification problems. Classification: A classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”. Regression: A regression problem is when the output variable is a real value, such as “dollars” or “weight”.



**Fig 4.2 Working of SVM Classifier**

Here in Fig 4.2 The  $H_1$  does not separate the classes.  $H_2$  does, but only with a small margin.  $H_3$  separates them with the maximal margin.

#### 4.1.2.1 Advantages of Supervised Learning:

Supervised learning allows you to collect data or produce a data output from the previous experience. Helps you to optimize performance criteria using experience. Supervised machine learning helps you to solve various types of real-world computation problems. Disadvantages of Supervised Learning: Decision boundary might be overtrained if your training set which doesn't have examples that you want to have in a class. You need to select lots of good examples from each class while you are training the classifier. Classifying big data can be a real challenge. Training for supervised learning needs a lot of computation time.

#### 4.1.3 Gaussian Mixture Model

GMM is also a type of clustering algorithm. As its name implies, each cluster is modelled according to a different Gaussian distribution. This flexible and probabilistic approach to

modelling the data means that rather than having hard assignments into clusters like k-

means, we have soft assignments. This means that each data point could have been generated by any of the distributions with a corresponding probability. In effect, each distribution has some ‘responsibility’ for generating a particular data point.

Gaussian Mixture Model background subtraction (GMM) method is nowadays used in many moving object detection applications. This common approach is performed statistically on each single pixel in the captured frames. Thus, it is well suitable for parallel processing.

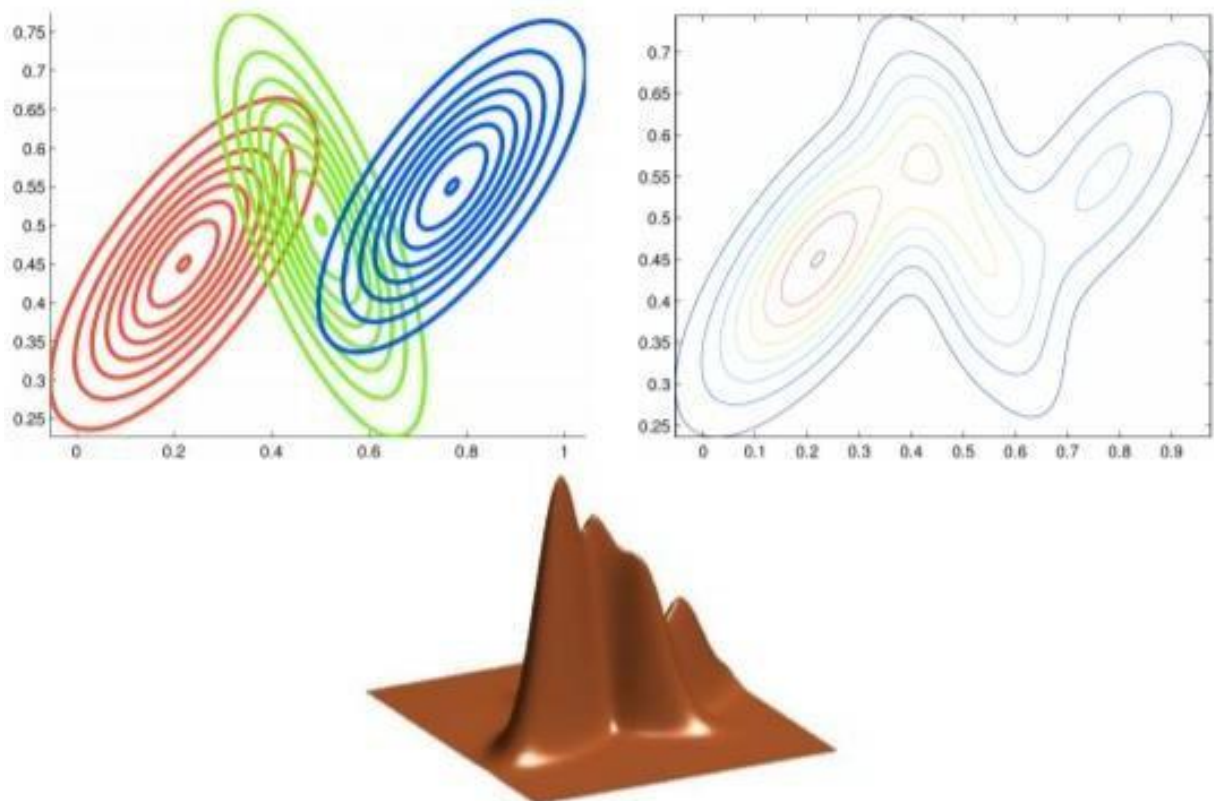
- A GMM represents a **distribution** as

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \mu_k, \Sigma_k)$$

with  $\pi_k$  the **mixing coefficients**, where:

$$\sum_{k=1}^K \pi_k = 1 \quad \text{and} \quad \pi_k \geq 0 \quad \forall k$$

**Fig 4.3 Formula's in GMM**



**Fig 4.4 Working process in GMM**

#### 4.1.4 HISTOGRAM OF ORIENTED GRADIENTS (HOG):

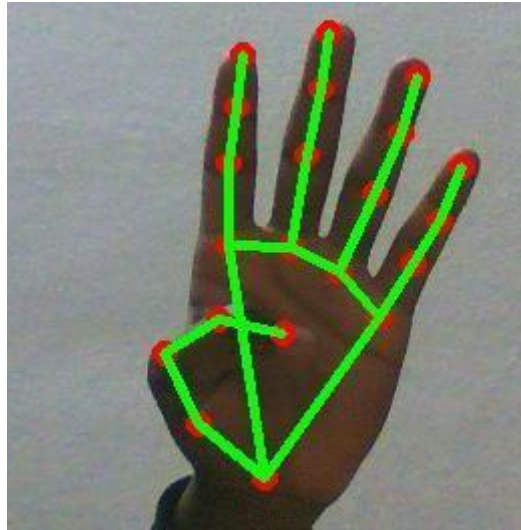
HOG, or Histogram of Oriented Gradients, is a feature descriptor that is often used to extract features from image data. It is widely used in computer vision tasks for object detection. Some important aspects of HOG that makes it different from other feature descriptors like Scale Invariant Feature Transform (SIFT), Speeded-Up Robust Feature (SURF) and many more:

1. The HOG descriptor focuses on the structure or the shape of an object. Now you might ask, how is this different from the edge features we extract for images? In the case of edge features, we only identify if the pixel is an edge or not. HOG is able to provide the edge direction as well. This is done by extracting the gradient and orientation (or you can say magnitude and direction) of the edges

2. Additionally, these orientations are calculated in 'localized' portions. This means that the complete image is broken down into smaller regions and for each region, the gradients and orientation are calculated. We will discuss this in much more detail in the upcoming sections

3. Finally the HOG would generate a Histogram for each of these regions separately. The histograms are created using the gradients and orientations of the pixel values, hence the name 'Histogram of Oriented Gradients'. HOG method is employed as a feature descriptor to acquire the face locations from the image, which is done by converting the given image into black and white and calculating the gradients (the direction in which the intensities of pixel increase) of each pixel in the image to extract the face data from the image. The magnitude and direction of gradient is calculated using the formula: The extracted data is then matched with an existing HOG face pattern to find the location of the face.

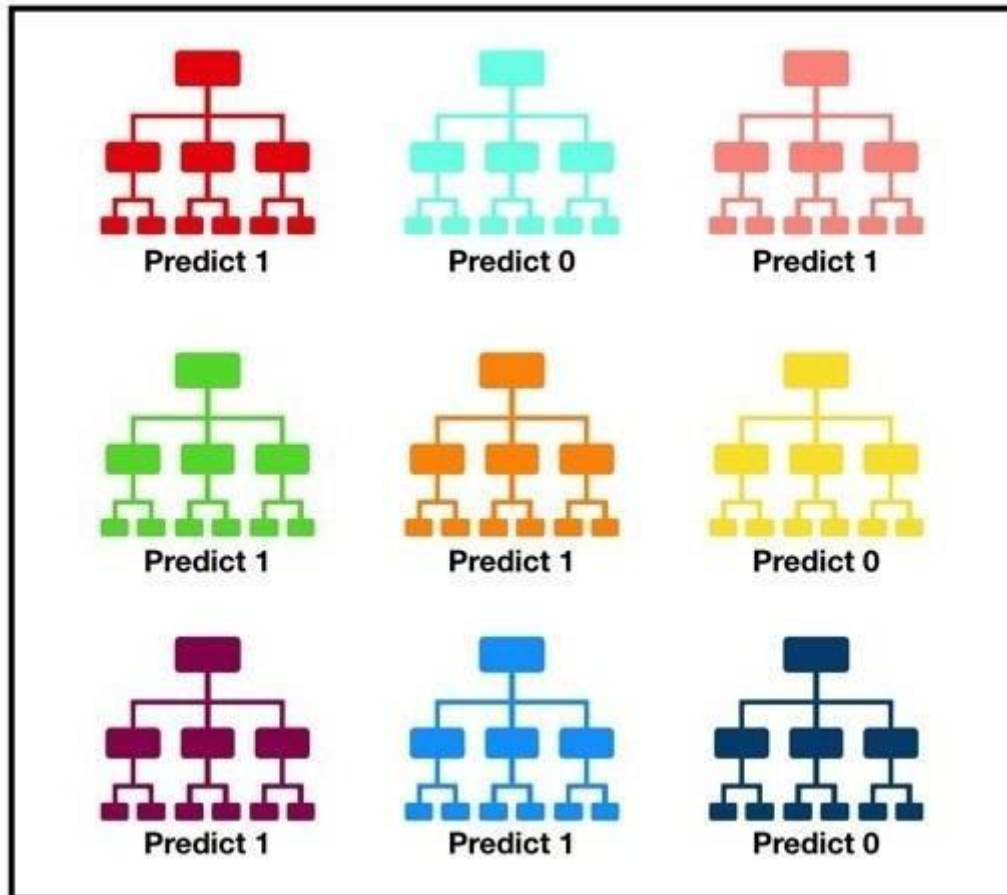
Fig 4.4, shows how the HOG face pattern is used to identify the face in the given image. This process is applied throughout the image to find all the faces present in the image. Fig 4.4 Face representation using HOG



**Fig 4.5 HOG recognises the Hand Gesture**

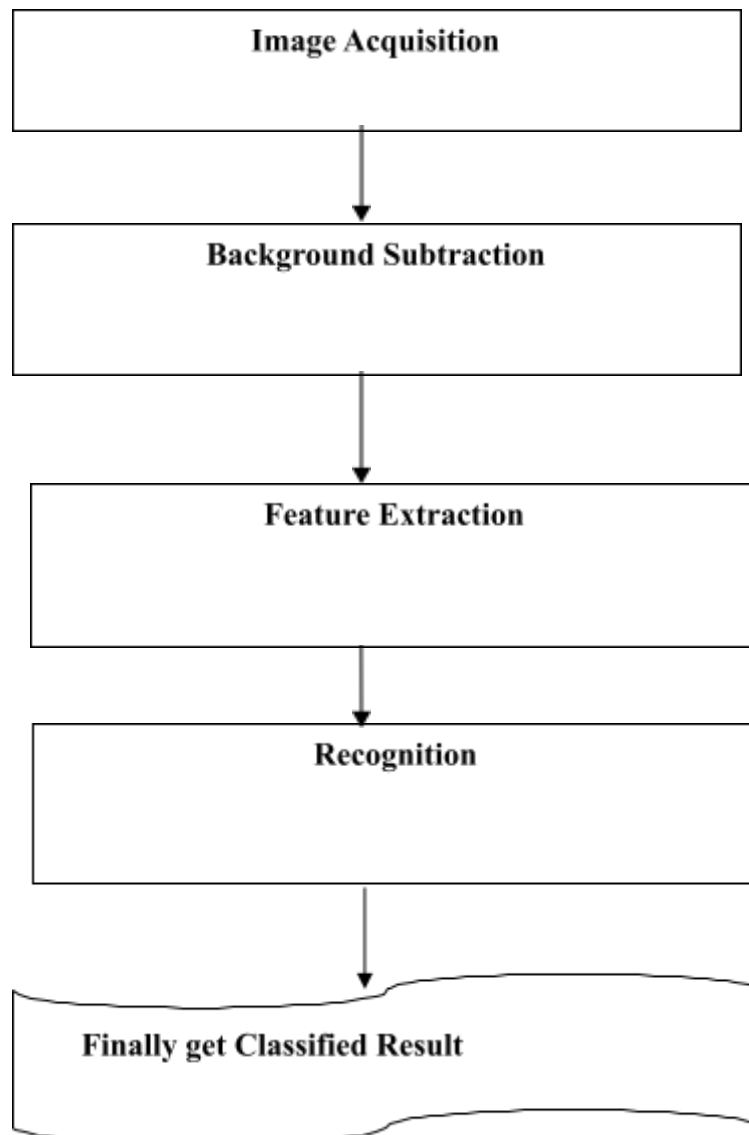
#### **4.1.5 Random Forest Classifier**

The Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction (see figure below). Visualization of a Random Forest Model Making a Prediction The fundamental concept behind random forest is a simple but powerful one — the wisdom of crowds. In data science speak, the reason that the random forest model works so well is: A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models. The low correlation between models is the key. Just like how investments with low correlations (like stocks and bonds) come together to form a portfolio that is greater than the sum of its parts, uncorrelated models can produce ensemble predictions that are more accurate than any of the individual predictions. The reason for this wonderful effect is that the trees protect each other from their individual errors (as long as they don't constantly all err in the same direction). While some trees may be wrong, many other trees will be right, so as a group the trees are able to move in the correct direction. So the prerequisites for random forest to perform well are: There needs to be some actual signal in our features so that models built using those features do better than random guessing. The predictions (and therefore the errors) made by the individual trees need to have low correlations with each other.



**Fig 4.6 Representation of Random Forest Classifier**

## 4.2 Proposed System



The proposed system has five different phases which involves 1. Image acquisition 2. Background Subtraction 3. Feature Extraction 4. Recognition 5. Classified Result. Image Acquisition is the phase that helps to capture the image using any device or a webcam Once the image is captured, the background disturbances are removed. Based on the image, the features are extracted using HOG method. The extracted features are recognized according to the training data and the classified result is displayed to the user.

## **4.3 Modules**

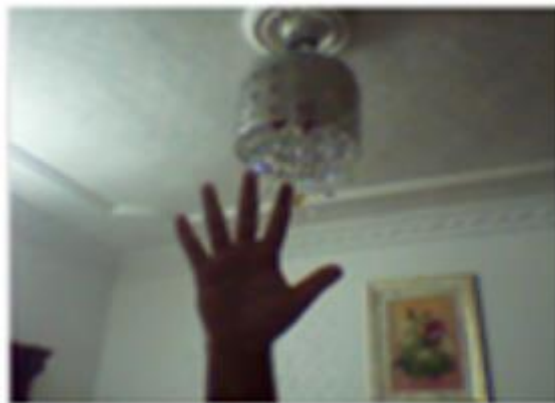
### **4.3.1 Background Subtraction**

The moving objects are isolated from the whole image while the static objects are considered as part from the background of the frame. This technique is implemented by creating a model for the background and updating it continuously to take into account light changes, shadows, overlapping of the objects in the visual area, or the new added static objects. This model is considered as a reference frame that will be subtracted from the current frame to detect the moving objects. There are many algorithms that can be used for background subtraction. This algorithm must have the ability to support multilevel illumination, detect all moving objects at different speeds, and consider any resident object as part of the background as soon as possible. Steps of background subtraction algorithms can be divided into four main stages, which are pre-processing, background modelling, foreground detection (referred to as background subtraction), and data validation (also known as postprocessing). The second stage is the modelling background stage which is also known as background maintenance. It is the main stage of any background subtraction algorithm.

The background modelling algorithms are classified into two main types: recursive and non-recursive models. Non recursive models use buffer to store all previous frames and estimate the background from the temporal variation of each pixel in the buffer. Recursive algorithms are not depending on the previous frames such as frame differencing, median filter, and linear predictive filter. Recursive models require less storage than non-recursive models. However, any error occurring in the recursive model can stay for longer time than non-recursive models.



**Fig 4.7 Step 1 in Background Subtraction**



**Fig 4.8 Step 2 in Background Subtraction**



**Fig 4.9 Step 3 in Background Subtraction**

### 4.3.2 Feature Extraction

In machine learning, pattern recognition and in image processing, feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. Feature extraction is related to dimensionality reduction. When the input data to an algorithm is too large to be processed and it is suspected to be redundant (e.g. the same measurement in both feet and meters, or the repetitiveness of images presented as pixels), then it can be transformed into a reduced set of features (also named a feature vector). Determining a subset of the initial features is called feature selection. The selected features are expected to contain the relevant information from the input data, so that the desired task can be performed by using this reduced representation instead of the complete initial data.

1. Feature extraction involves reducing the number of resources required to describe a large set of data. When performing analysis of complex data one of the major problem's stems from the number of variables involved. Analysis with a large number of variables generally requires a large amount of memory and computation power, also it may cause a classification algorithm to overfit to training samples and generalize poorly to new samples. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy. Many machine learning practitioners believe that properly optimized feature extraction is the key to effective model construction.
2. Results can be improved using constructed sets of application-dependent features, typically built by an expert. One such process is called feature engineering. Alternatively, general dimensionality reduction techniques are used such as:
3. Independent component analysis
4. Isomap
5. Image processing

One very important area of application is image processing, in which algorithms are used to

detect and isolate various desired portions or shapes (features) of a digitized image or video stream. It is particularly important in the area of optical character recognition.

1. Low-level
2. Edge detection
3. Corner detection
4. Blob detection
5. Active contours
6. Feature extraction in software

Many data analysis software packages provide for feature extraction and dimension reduction. Common numerical programming environments such as MATLAB, SciLab, NumPy and the R language provide some of the simpler feature extraction techniques (e.g. principal component analysis) via built-in commands. More specific algorithms are often available as publicly available scripts or third-party add-ons. There are also software packages targeting specific machine learning applications that specialize in feature extraction.

#### **4.3.3 CLASSIFICATION**

Decision Tree learning uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modelling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data (but the resulting classification tree can be an input for decision making). Decision tree learning is the construction of a decision tree from class-labeled training tuples. A decision tree is a flow-chart-like structure, where each internal (non-leaf) node denotes a test on an attribute, each branch represents the outcome of a test, and each leaf (or terminal) node holds a class label. The topmost node in a tree is the root node.

Decision trees used in data mining are of two main types:

Classification tree analysis is when the predicted outcome is the class to which the data belongs. Regression tree analysis is when the predicted outcome can be considered a real

number (e.g. the price of a house, or a patient's length of stay in a hospital). The term Classification and Regression Tree (CART) analysis is an umbrella term used to refer to both of the above procedures, first introduced by Breiman et al. Trees used for regression and trees used for classification have some similarities – but also some differences, such as the procedure used to determine where to split.

Algorithms for constructing decision trees usually work top-down, by choosing a variable at each step that best splits the set of items.<sup>[14]</sup> Different algorithms use different metrics for measuring “best”. These generally measure the homogeneity of the target variable within the subsets. Some examples are given below. These metrics are applied to each candidate subset, and the resulting values are combined (e.g., averaged) to provide a measure of the quality of the split.

Above-generated feature vector for each image is trained through by classifiers like Random Forest, and finally that will be the last technique. Accuracy is calculated for model and by passing them after the testing set. Random Forest, the popular most model in supervised machine learning models. RF, an ensemble model which works for regression and classification. So, it is called as ensemble Random forest that is a model of integrating several classifiers ordered to solve complex problems and enhance efficiency of the model. In RF, before having result multiple decision trees used. Thus, combining many trees for decision.

The working can be illustrated in the below steps and diagram:

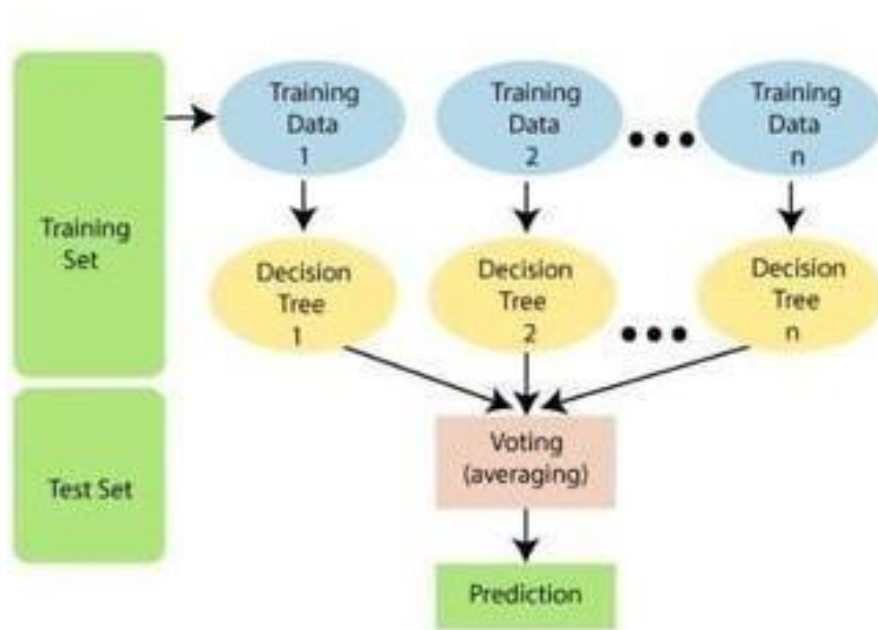
Step 1: From training set, pick random  $P$  data points.

Step 2: Form decision trees correlated with the chosen data points (subsets).

Step 3: Denote the number of decision trees to build as  $N$ .

Step 4: Repeat initial two steps.

Step 5: Appraise the result of every decision tree with fresh data points and allocate them to a group-based on majority votes



**Fig 4.10 Working of Decision Tree**

## CHAPTER 5

### RESULTS AND DISCUSSION

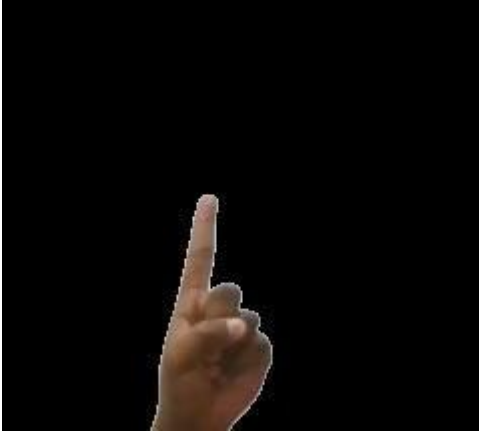
1. The system uses image processing and machine learning technologies such as HOG, GMM, Random Forest and SVM to identify hand gestures such as alphabets, numbers and words to predict the precise outcome of the hand sign and establish stable communication to a speech-impaired individual.
2. For training the dataset 300 hand gesture images were used, which were acquired from 8 different individuals to be able to recognize the sign better through each person's hand nuances.
3. The system was tested with 50 images and it was successfully able to recognize 8 unique gestures with 94.8% accuracy.
4. Since sign language involves hand gestures with their motion in 3D space in correspondence to other features such face, the complete Indian sign language cannot be implemented in this method. But the gestures essential for basic communication, can be established through this system.
5. This system can be used by a speech impaired individual to be able to achieve day-to-day basic tasks.



**Fig 5.1 - Dataset Sample for "Call"**



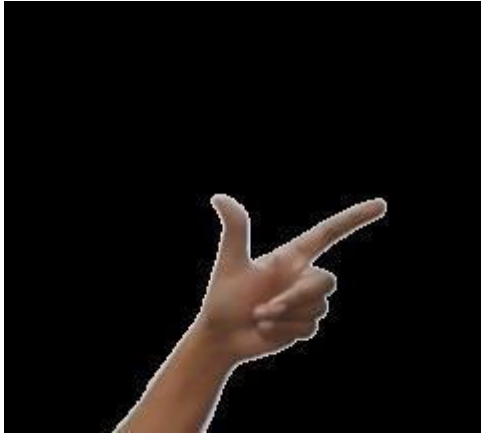
**Fig 5.2 - Dataset Sample for "Go"**



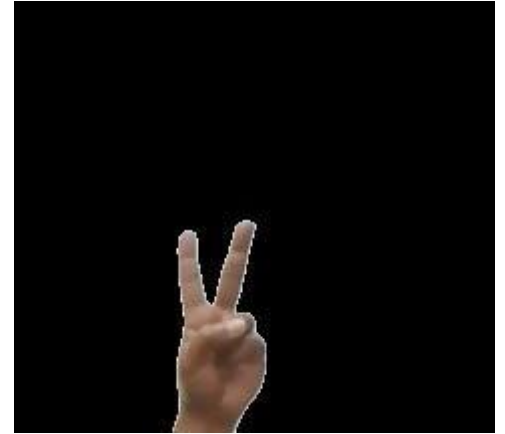
**Fig 5.3 Dataset Sample for “One”**



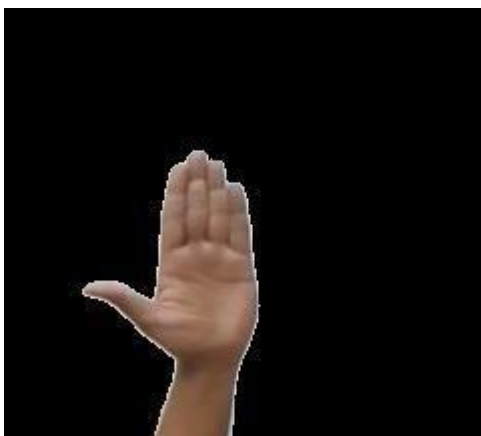
**Fig 5.4 Dataset Sample for “Five”**



**Fig 5.5 Dataset Sample for “Looser”**



**Fig 5.6 Dataset Sample for”Two”**

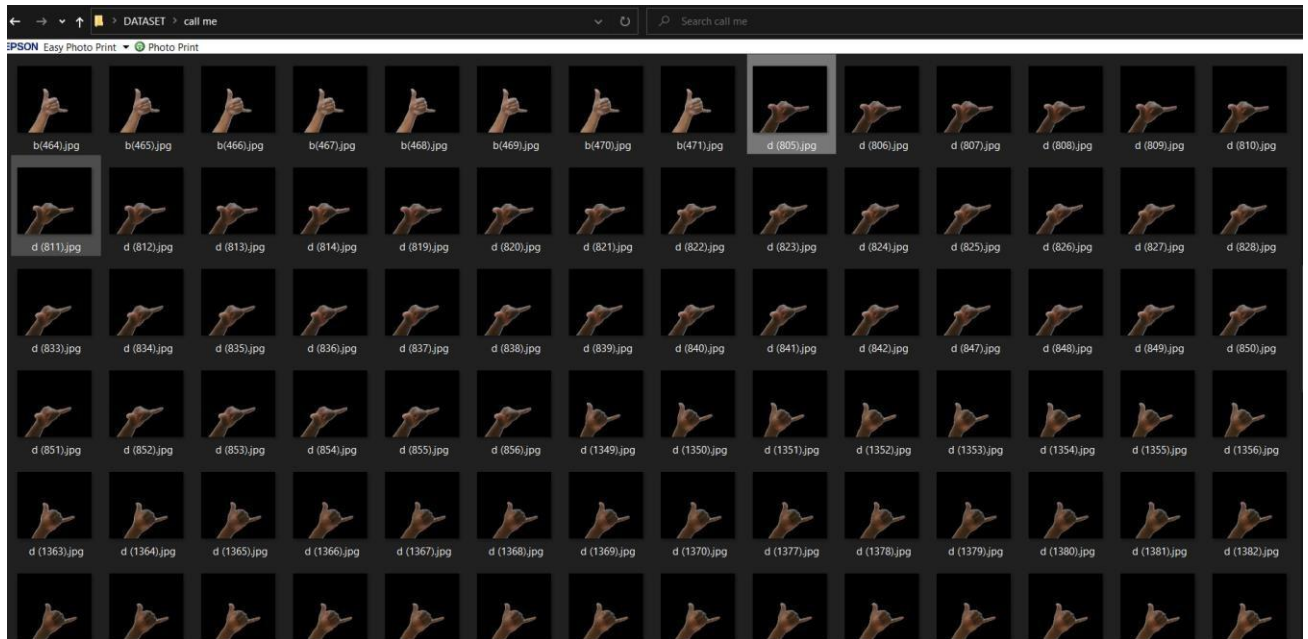


**Fig 5.7 Dataset Sample for “Stop”**

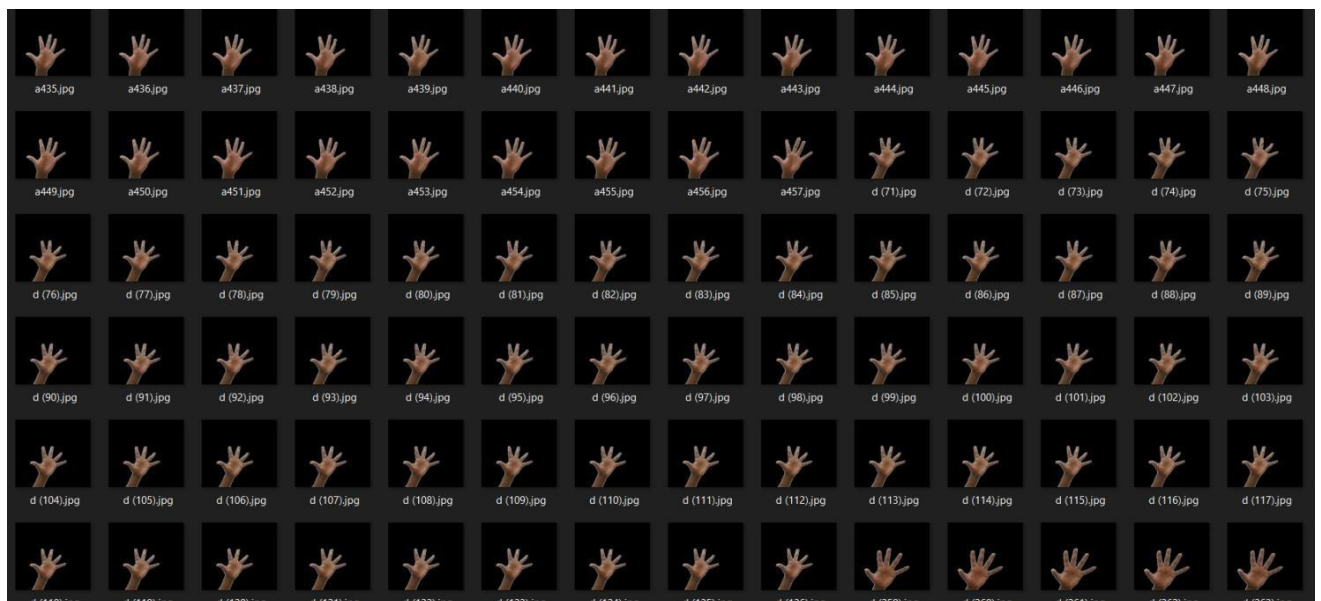


**Fig 5.8 Dataset Sample  
for “Thumbs Up”**

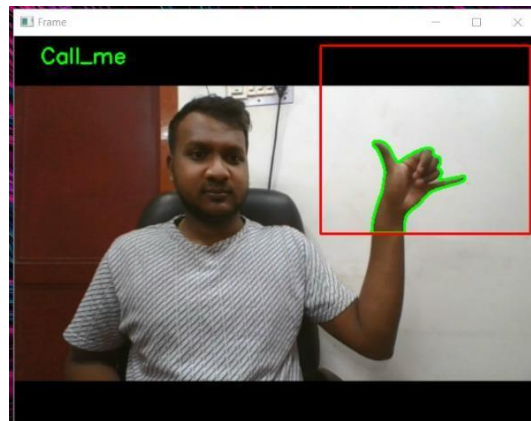
Dataset for each hand gesture was taken from 8 individuals. Its accuracy will get even more high if it was taken from many more people but having Covid restrictions in mind this was the maximum we could achieve. Almost 250 to 300 sample photos of a single hand gesture were used in order to train the model and thereby achieve maximum results.



**Fig 5.9 Complete data set of call me**



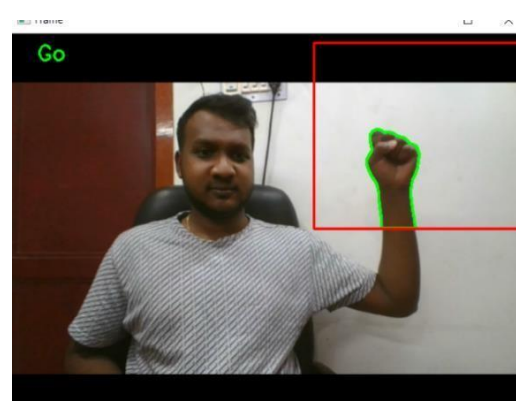
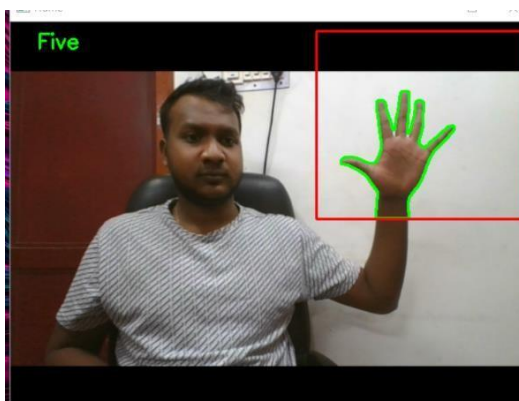
**Fig 5.10 Complete data set of five**



**Fig 5.11 Final Output of Call me**

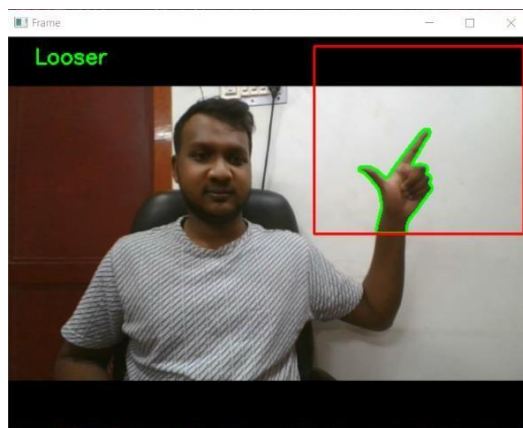
#### **Steps to use the application:**

- Open the application
- The main window is displayed which accesses the camera and displays the viewfinder.
- A portion of the viewfinder highlighted and this portion will be used to recognize the hand gestures.
- Initially the application takes snapshots of the gesture recognition portion without the hand, these snapshots will be used for background subtraction.
- The user can now place the hand in this portion and show the hand sign.
- This gesture will be captured, processed, recognized and its corresponding word or number will be displayed in the top-left portion of the viewfinder.
- And the other individual can respond accordingly, thereby facilitating a relatively quick and simple communication process.

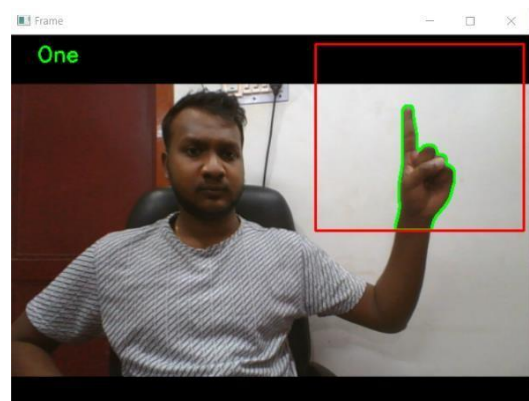


**Fig 5.12 – Final Output of “Five”**

**Fig 5.13 – Final Output of “Stop”**



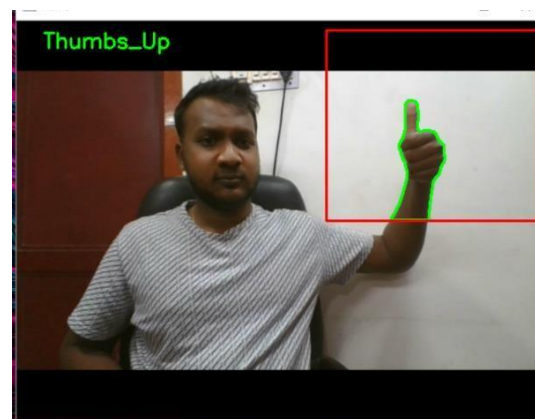
**Fig 5.14 Final Output of “Looser”**



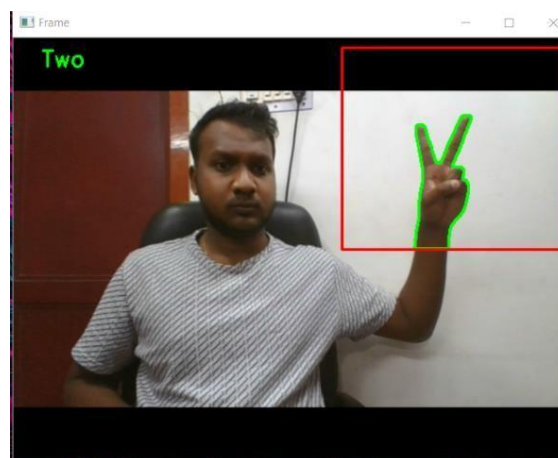
**Fig 5.15 Final Output of “One”**



**Fig 5.16 Final Output of “Stop”**



**Fig 5.17 Final Output of “Thumbs Up”**



**Fig 5.18 Final Output of “Two”**

## **CHAPTER 6**

### **CONCLUSION AND FUTURE WORK**

#### **6.1 CONCLUSION**

In this paper, we used novel methodology to present automated hand sign language gesture recognition in real time. Sign language recognition system was successfully implemented by hand detection using skin color segmentation, shape feature extraction using HOG and classification using machine learning model i.e., Random Forest. The proposed model obtains the highest accuracy like 94.3% by HOG and Random Forest classifiers mixed. This work will be extended in the future to enhance obtained results and create a Sign-Voice application that helps people with listening impairments to converse more fluently with people.

#### **6.2 FUTURE WORK**

In future, we can add voice to the application so that people who doesn't even know to read can hear the audio of that particular gesture and they can easily help the impaired people. Sign language uses both hands for showing hand gestures, and moreover these hand gestures aren't stationary. These gestures are basically actions with motion. So, its hard to capture the particular hand gesture with a single camera. We will need cameras in 3-D space so that each gesture movements can be captured or recorded and then give the users the proper response, So these changes can be made and thereby developing the project.

## REFERENCES

- [1] JOYEETA SINGH, K. D. "Indian sign language recognition using eigen value weighted euclidean distance-based classification technique", International Journal of Advanced Computer Science and Applications, 2013.
- [2] NEHA V. TAVARI, P. A. V. Debrancher. "Indian sign language recognition based on histograms of oriented gradient", International Journal of Computer Science and Information Technologies Vol 5 (3), 2014.
- [3] NMANIVAS. Gesture recognition system  
<https://github.com/nmanivas/GestureRecognitionSystem>  
.
- [4] PADMAVATHI S, SAIPREETHY.M.S, V. "Indian sign language character recognition using neural networks", IJCA Special Issue on Recent Trends in Pattern Recognition and Image Analysis, RTPRIA, 2013.
- [5] SAKSHI GOYAL, ISHITA SHARMA, S. S. "Sign language recognition system for deaf and dumb people", International Journal of Engineering Research Technology, April 2013.
- [6] Yiwen He, Jianyu Yang, Zhanpeng Shao, Youfu Li, "Salient feature point selection for real time RGB-D hand gesture recognition", IEE International Conference on real- time Computing and Robotics, 2017.
- [7] Rania A. Elsayed, Mahmoud I. Abdalla, Mohammed S. Sayed, "Hand gesture recognition based on dimensionality reduction of histogram of oriented gradients", Japan-Africa Conference on Electronics, Communication and Computers, 2017.
- [8] HimadriNathSaha, SayanTapadar, Shinjini Ray, Suhrid Krishna Chatterjee, "A Machine Learning based approach for Hand Gesture Recognition using distinctive

feature extraction",IEEE 8th Annual Conference on Computing and Communication,  
2018

- [9] Mukesh Kumar Makwana, " Sign Language Recognition", Indian Institute of Science, Bangalore.
- [10] Escalera, Sergio, et al. "Chalearn looking at people challenge 2014: Dataset and results", Workshop at the European Conference on Computer Vision, Springer International Publishing, 2014.
- [11] R. Saxena, S. Hussain, X. Han, H. Shin and J. A. Khan, "Hand gesture recognition using deep learning", International SoC Design Conference (ISOCC) Seoul, 2017.
- [12] Takayoshi Yamashita and Taro Watasue, "Hand posture recognition based on bottom-up structured deep convolutional neural network with curriculum learning", IEEE International Conference on Image Processing (ICIP) Paris, 2014.

## APPENDIX

### A. Source Code

```
from tool.gesture_recognition import MotionDetector
import numpy as np
import imutils
import cv2
import os
```

```
folder = 'm'
```

```
try :
```

```
    os.mkdir(folder)
```

```
except FileExistsError:
```

```
    pass
```

```
Alphabet = folder
```

```
camera = cv2.VideoCapture(0)
```

```
ROI = "10,350,225,590"
```

```
(top, right, bot, left) = np.int32(ROI.split(","))
```

```
md = MotionDetector()
```

```
numFrames = 0
```

```
k = 0
```

```
# keep looping
```

```
while True:
```

```
    # grab the current frame
```

```
(grabbed, frame) =  
camera.read()
```

```

# resize the frame and flip it so the frame is no longer a mirror view
frame = imutils.resize(frame, width=600)
frame = cv2.flip(frame, 1)
clone = frame.copy()
(frameH, frameW) = frame.shape[:2]

# extract the ROI, passing in right:left since the image is mirrored,
then # blur it slightly
roi = frame[top:bot,
right:left] hand = roi.copy()
gray = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)
gray = cv2.GaussianBlur(gray, (3, 3), 0)

# if we not reached 32 initial frames, then calibrate the skin detector
if numFrames < 100:
    md.update(gray)

# otherwise, detect skin in the
ROI else:
    # detect motion (i.e., skin) in the image
    skin = md.detect(gray)

# check to see if skin has been
detected if skin is not None:
    # unpack the tuple and draw the contours on the
    image (thresh, c) = skin
    masked = cv2.bitwise_and(hand, hand, mask=thresh)

```

```
cv2.imshow("Mask", masked)
```

```
name = folder + "\\ " + Alphabet + str(k) + ".jpg"
cv2.imwrite(name, masked)
cv2.drawContours(clone, [c + (right, top)], -1, (0, 255, 0),
2) cv2.imshow("Thresh", thresh)
k = k+1
```

# draw the hand ROI and increment the number of processed frames

```
cv2.rectangle(clone, (left, top), (right, bot), (0, 0, 255), 2)
```

```
numFrames += 1
```

```
if numFrames >= 100:
```

```
    if fl == 1:
```

```
        print ("Calibration Completed")
```

```
        fl = 0
```

```
else :
```

```
    print
```

```
    (numFrames) fl =
```

```
    1
```

# show the frame to our

```
screen cv2.imshow("Frame",
```

```
clone) key = cv2.waitKey(1) &
```

```
0xFF
```

# if the 'q' key is pressed, stop the

```
loop if key == ord("q"):
```

```
    break
```

```
# cleanup the camera and close any open windows
camera.release()
cv2.destroyAllWindows()
import argparse
```

```

import cv2

from imutils import paths

import imutils

# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--image", required=True, help="Path to the image")
args = vars(ap.parse_args())

# load the image, convert it to grayscale, and blur it slightly
for imagePath in paths.list_images(args["image"]):
    image = cv2.imread(imagePath)

    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    blurred = cv2.GaussianBlur(gray, (7, 7), 0)
    #cv2.imshow("Image", image)

    # apply basic thresholding -- the first parameter is the image
    # we want to threshold, the second value is is our threshold
    # check; if a pixel value is greater than our threshold (in this
    # case, 200), we set it to be BLACK, otherwise it is WHITE.
    (T, threshInv) = cv2.threshold(blurred, 15, 255, cv2.THRESH_BINARY)
    #cv2.imshow("Threshold Binary Inverse", threshInv)

    cnts = cv2.findContours(threshInv.copy(),
cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)

    cnts = cnts[0] if imutils.is_cv2() else
cnts[1]
    c = max(cnts,
key=cv2.contourArea)

```

```
(x, y, w, h) = cv2.boundingRect(c)
crop = threshInv[y:y+h,x:x+w]
crop_resized = cv2.resize(image, (100, 100))
```

```
cv2.imwrite(imagePath,crop_resized)
print (imagePath)
```

```
# import the necessary packages
from skimage import feature
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.ensemble import RandomForestClassifier
from imutils import paths
import cv2
import numpy as np
import pickle as cPickle
```

```
data = []
labels = []
```

```
# loop over the training images
for imagePath in paths.list_images("DATASET"):
    # load the image, convert it to grayscale, and describe
    it image = cv2.imread(imagePath,0)
    print (imagePath)
    #gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    H = feature.hog(image, orientations=9, pixels_per_cell=(10,
10), cells_per_block=(2, 2), transform_sqrt=True)

    labels.append(imagePath.split("\\")[-2])
    data.append(H)
```

```
(trainData, testData, trainLabels, testLabels) = train_test_split(np.array(data),
```

```

np.array(labels), test_size=0.25, random_state=42)

# train a random forest on the data
#model = LinearSVC(C=100.0, random_state=42)
print("[INFO] training model...")
model = RandomForestClassifier(n_estimators=20, random_state=42)
model.fit(data, labels)
print("[INFO] evaluating...")
predictions = model.predict(testData)
print(classification_report(testLabels, predictions))
f = open("classifier.cPickle", "wb")
f.write(cPickle.dumps(model))
f.close()

# import the necessary packages
from tool.gesture_recognition import MotionDetector
import numpy as np
import imutils
import cv2
import os

from sklearn.ensemble import RandomForestClassifier
from skimage import exposure
from skimage import feature
#import mahotas
import cv2
import pickle as cPickle

camera = cv2.VideoCapture(0)

```

ROI = "10,350,225,590"

```

# unpack the hand ROI, then initialize the motion detector and the total
number of

# frames read thus far
(top, right, bot, left) = np.int32(ROI.split(", "))
md = MotionDetector()
numFrames = 0

#model = RandomForestClassifier(n_estimators=20, random_state=42)
loaded_model = cPickle.load(open("classifier.cPickle", 'rb'))


# keep looping
while True:

    # grab the current frame
    (grabbed, frame) = camera.read()

    frame = cv2.flip(frame, 1)

    # resize the frame and flip it so the frame is no longer a mirror view
    frame = imutils.resize(frame, width=600)

    #frame = cv2.flip(frame, 1)
    clone = frame.copy()

    (frameH, frameW) = frame.shape[:2]


    # extract the ROI, passing in right:left since the image is mirrored,
    then # blur it slightly
    roi = frame[top:bot,
    right:left] hand = roi.copy()

    gray = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)
    gray = cv2.GaussianBlur(gray, (7, 7), 0)

```

# if we not reached 32 initial frames, then calibrate the skin detector

```

if numFrames< 32:
    md.update(gray)

# otherwise, detect skin in the
ROI else:
    # detect motion (i.e., skin) in the image
    skin = md.detect(gray)

# check to see if skin has been
detected if skin is not None:
    # unpack the tuple and draw the contours on the image
    (thresh, c) = skin
    #masked = cv2.bitwise_and(hand, hand, mask=thresh)
    #gray = cv2.cvtColor(masked, cv2.COLOR_BGR2GRAY)
    #edged = imutils.auto_canny(gray)
    (_,cnts, _) =
cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    c = max(cnts, key=cv2.contourArea)
    (x, y, w, h) = cv2.boundingRect(c)
    logo = thresh[y:y + h, x:x + w]
    cv2.imshow("logo",logo)
    logo = cv2.resize(logo, (100, 100))
    H = feature.hog(logo, orientations=9, pixels_per_cell=(10, 10),
cells_per_block=(2, 2), transform_sqrt=True)
    pred = loaded_model.predict(H.reshape(1, -1))[0]
    #prediction = loaded_model.predict(features.reshape(1,
-1))[0] print (pred)

```

```
cv2.putText(clone, pred, (30, 30),  
cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 255, 0), 2)
```

```
#cv2.imshow("Mask", masked)
cv2.drawContours(clone, [c + (right, top)], -1, (0, 255, 0),
2) cv2.imshow("Thresh", thresh)
```

```
# draw the hand ROI and increment the number of processed frames
```

```
cv2.rectangle(clone, (left, top), (right, bot), (0, 0, 255), 2)
```

```
numFrames += 1
```

```
if numFrames >= 30:
```

```
    if fl == 1:
```

```
        print ("Calibration Completed")
```

```
        fl=0
```

```
else :
```

```
    print
```

```
    (numFrames) fl =
```

```
    1
```

```
# show the frame to our
```

```
screen cv2.imshow("Frame",
```

```
clone) key = cv2.waitKey(1) &
```

```
0xFF
```

```
# if the 'q' key is pressed, stop the
```

```
loop if key == ord("q"):
```

```
    break
```

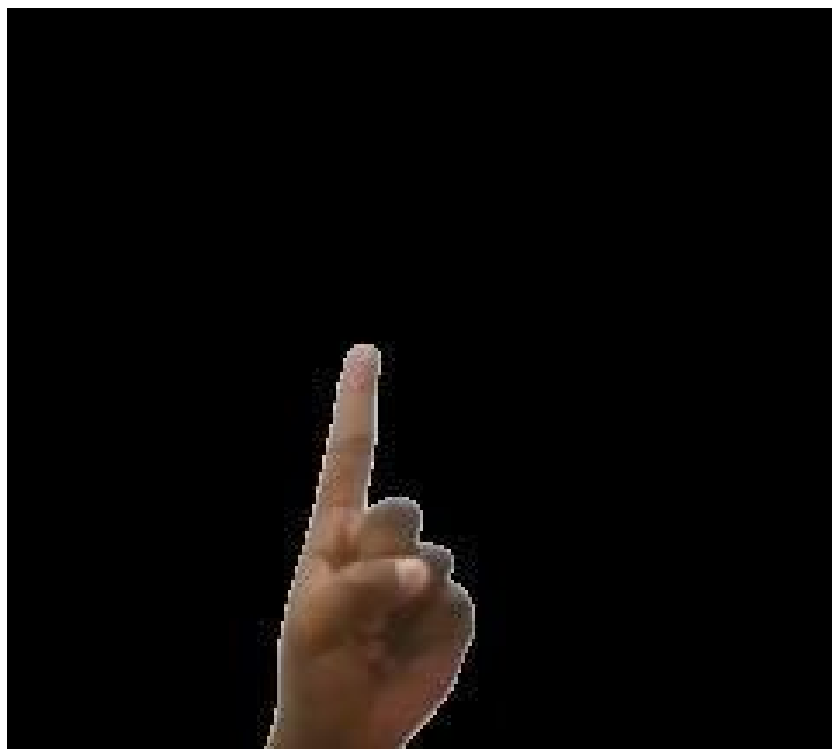
```
# cleanup the camera and close any open windows
```

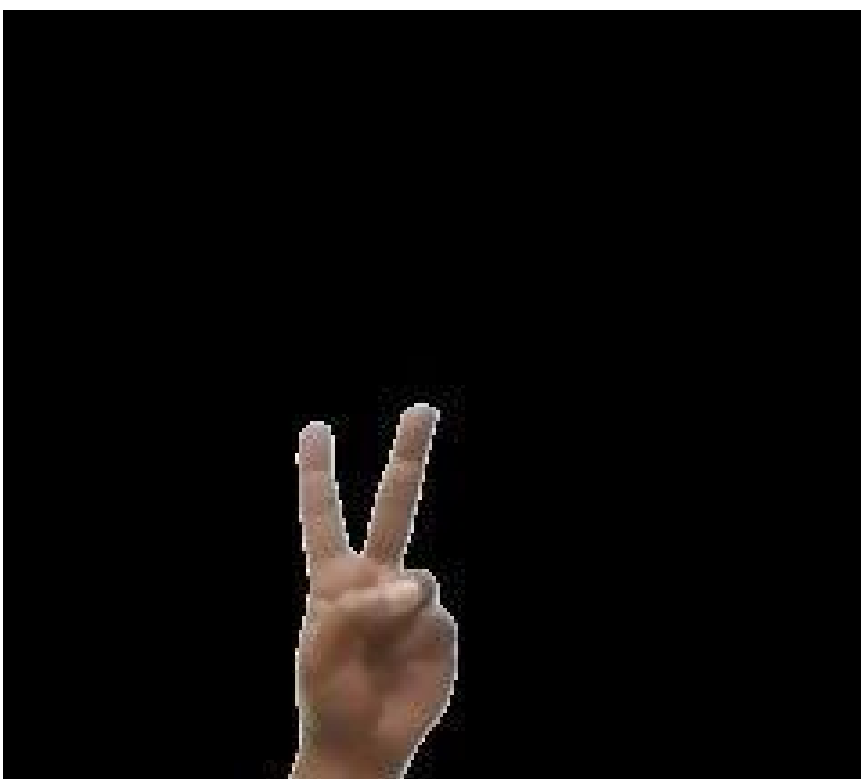
```
camera.release()
```

```
cv2.destroyAllWindows()
```

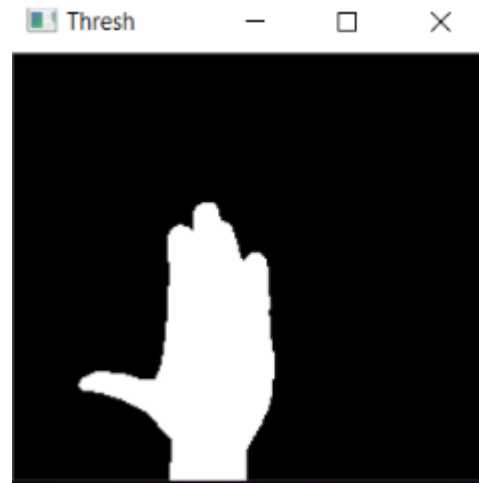
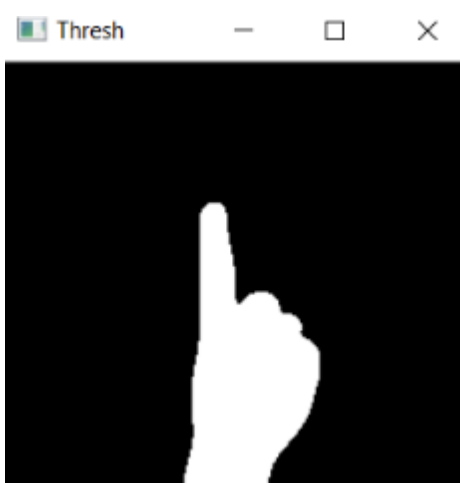
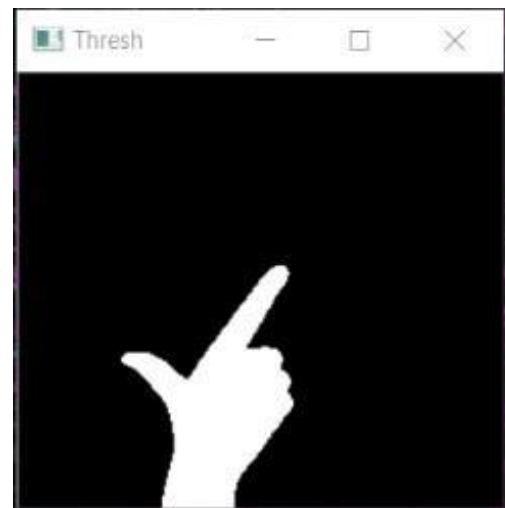
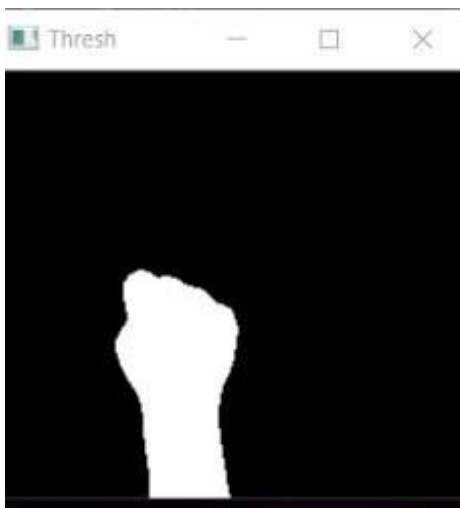
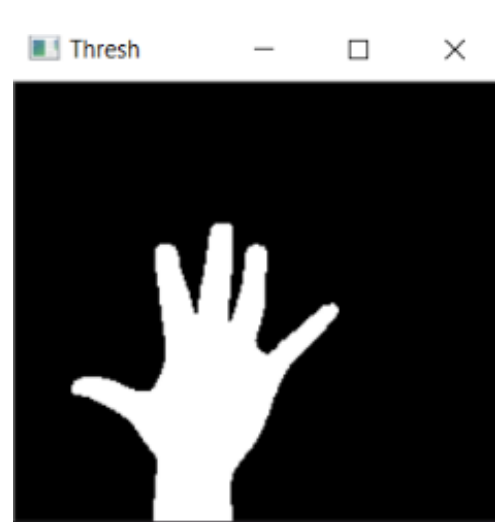
## B. SCREENSHOTS

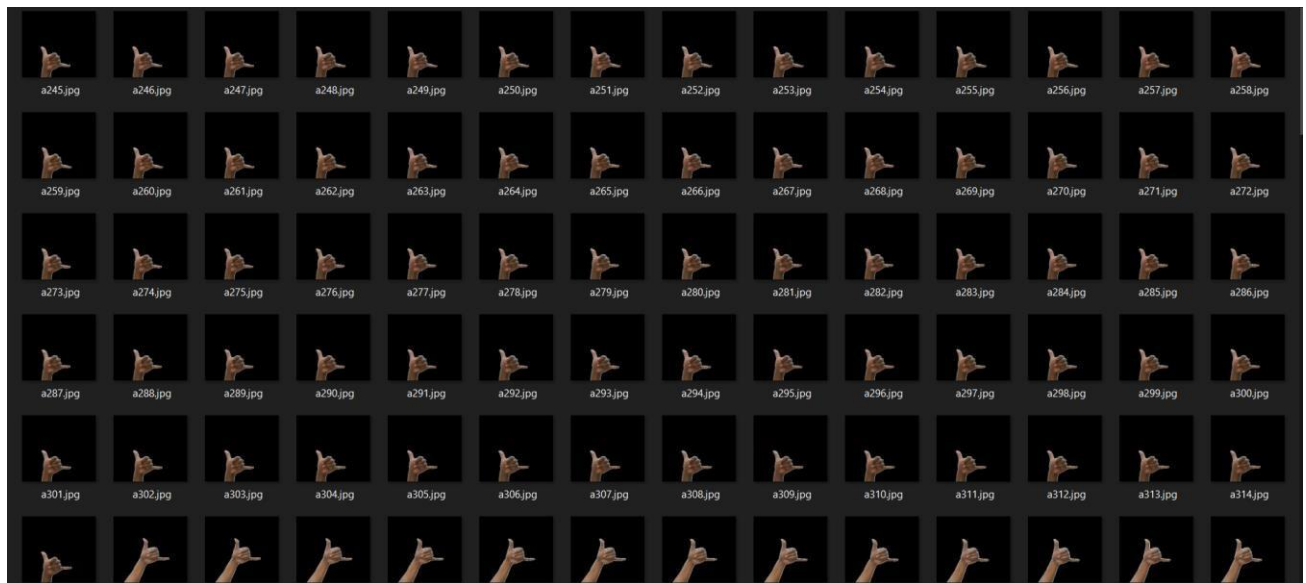
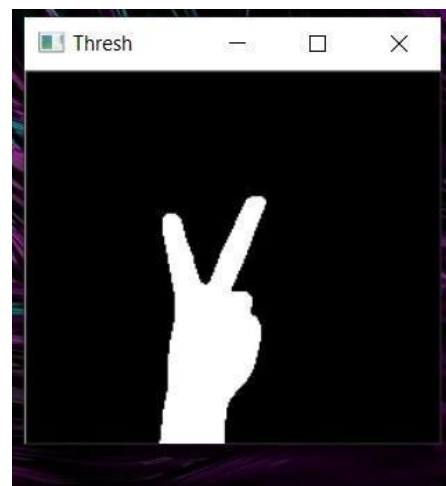




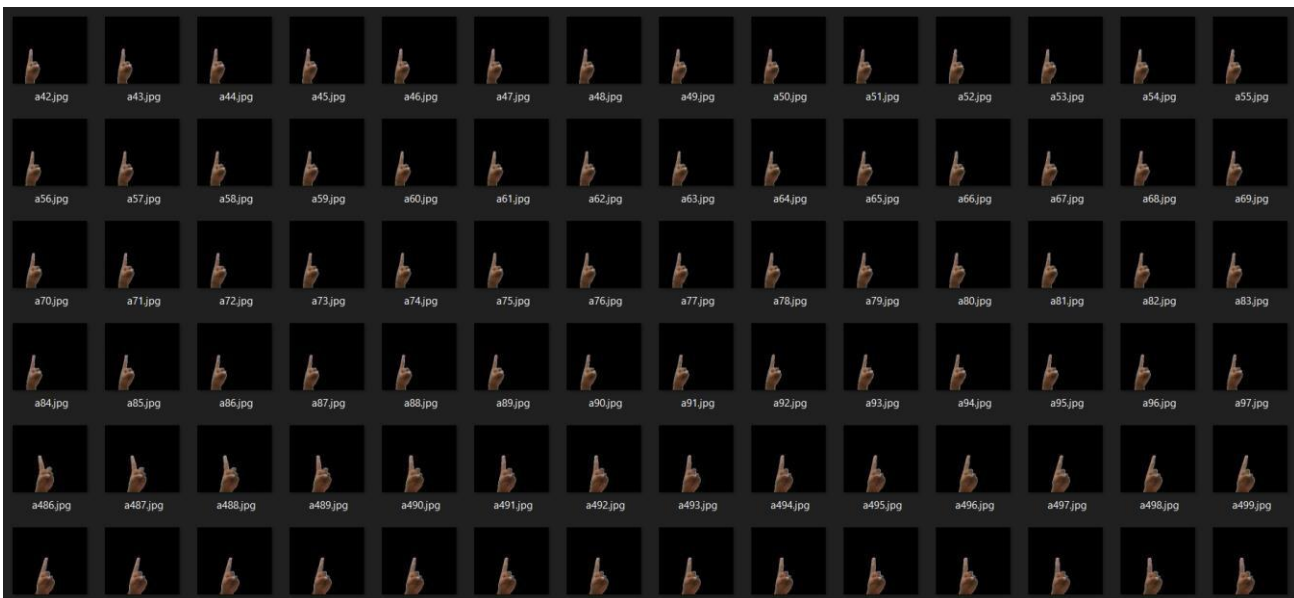
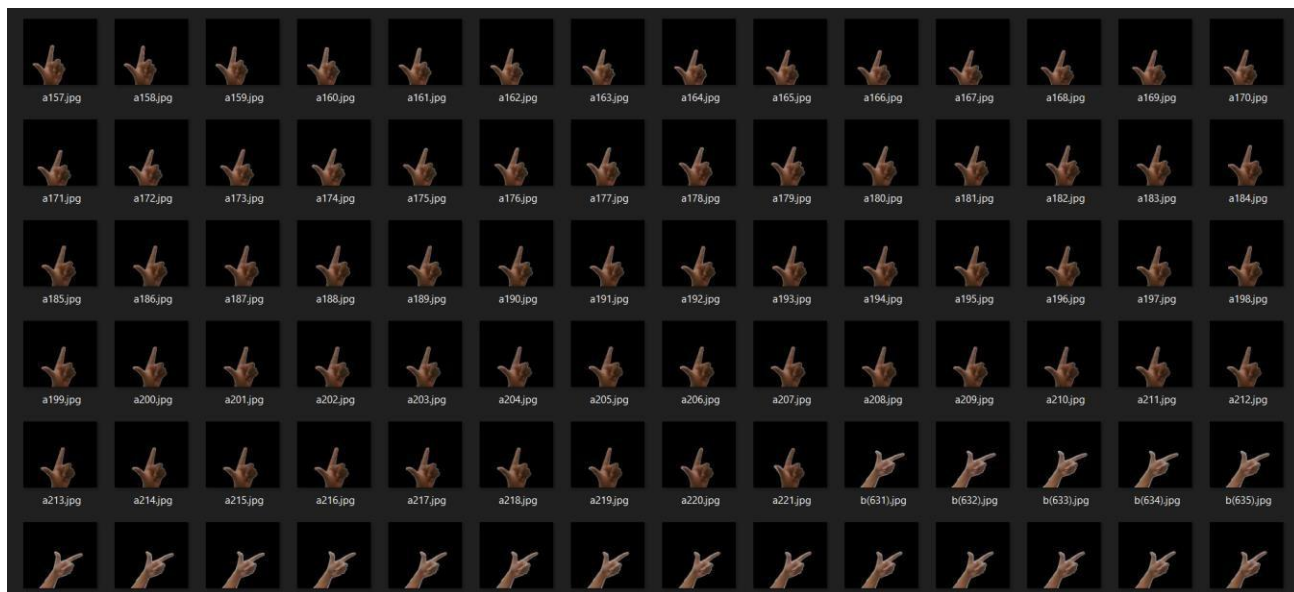




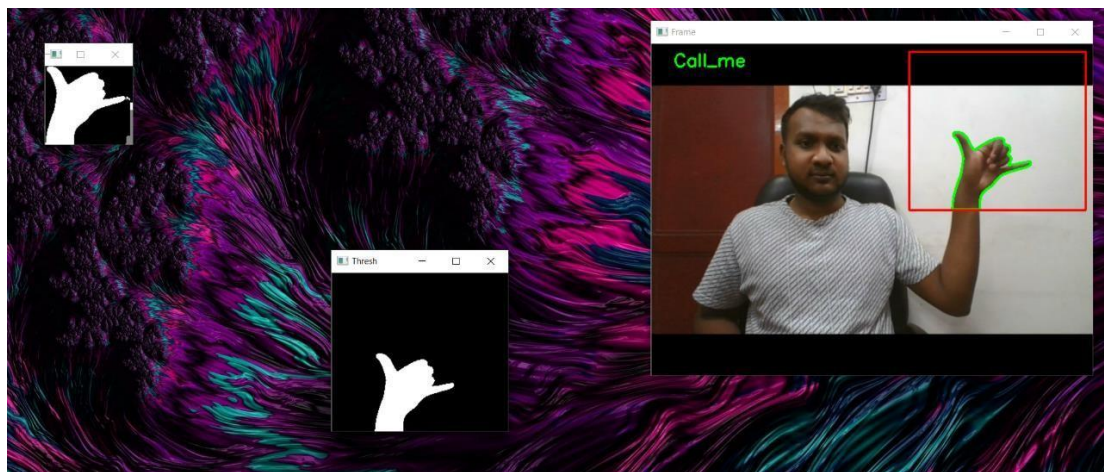
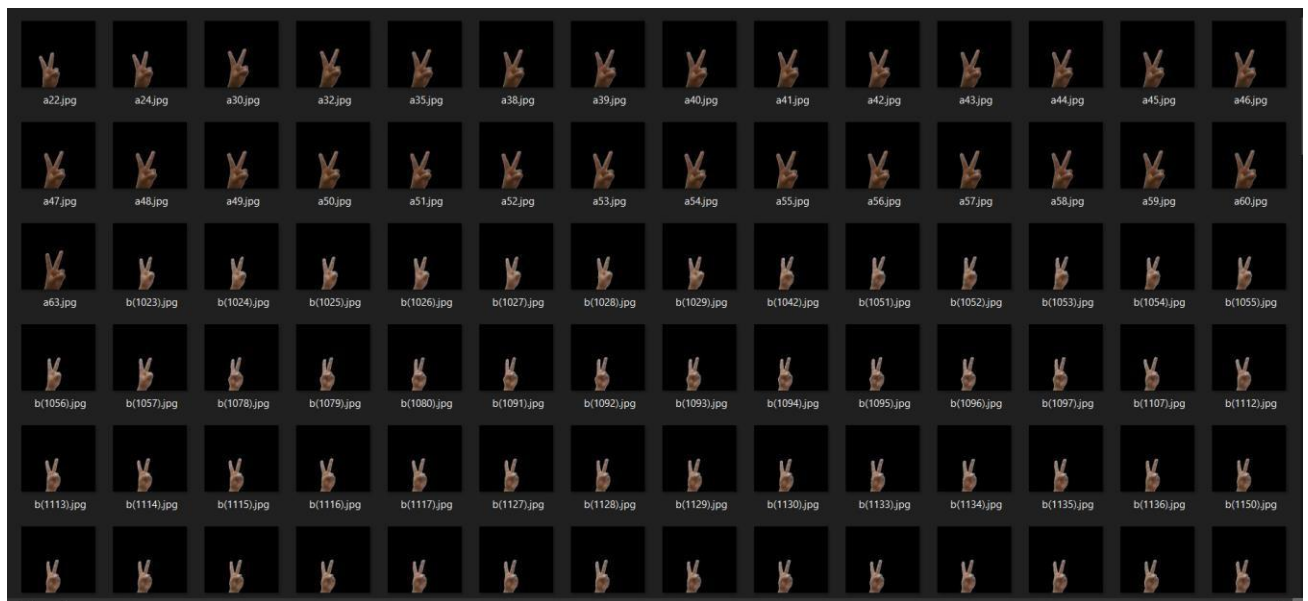


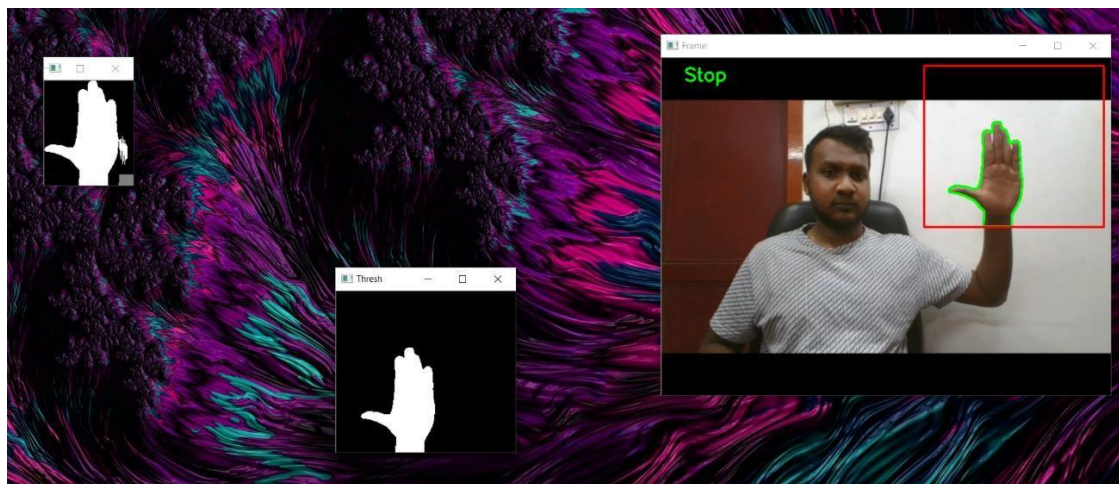
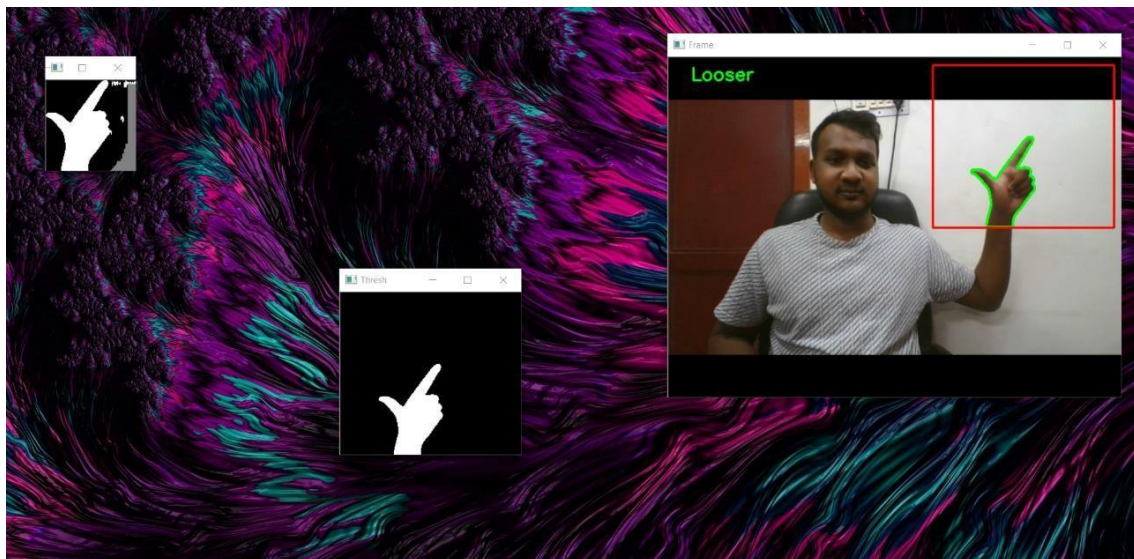
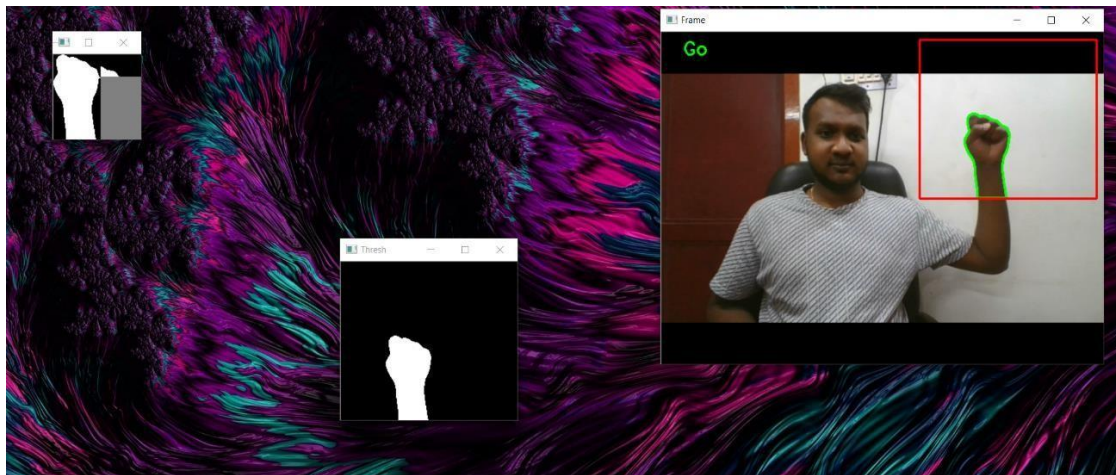


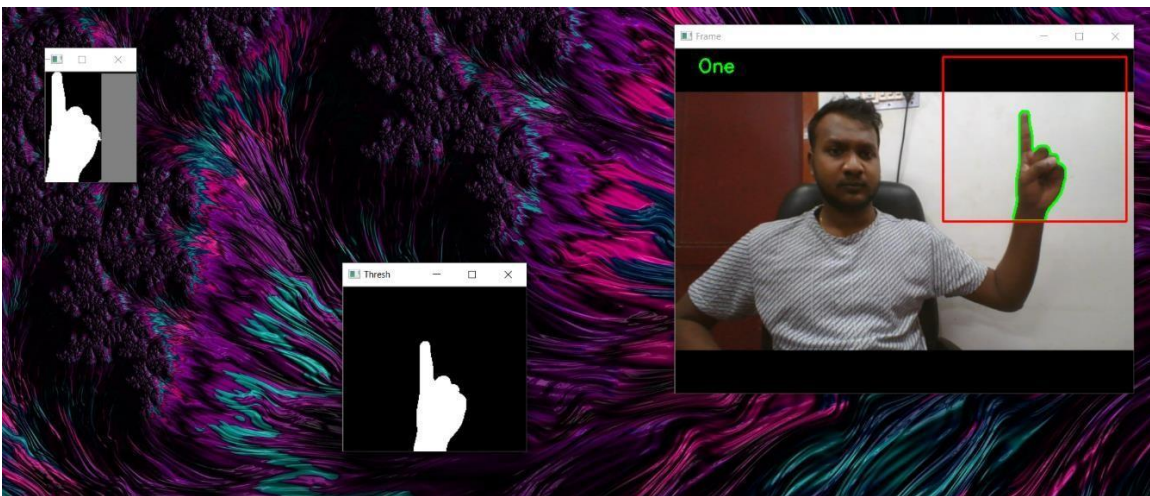
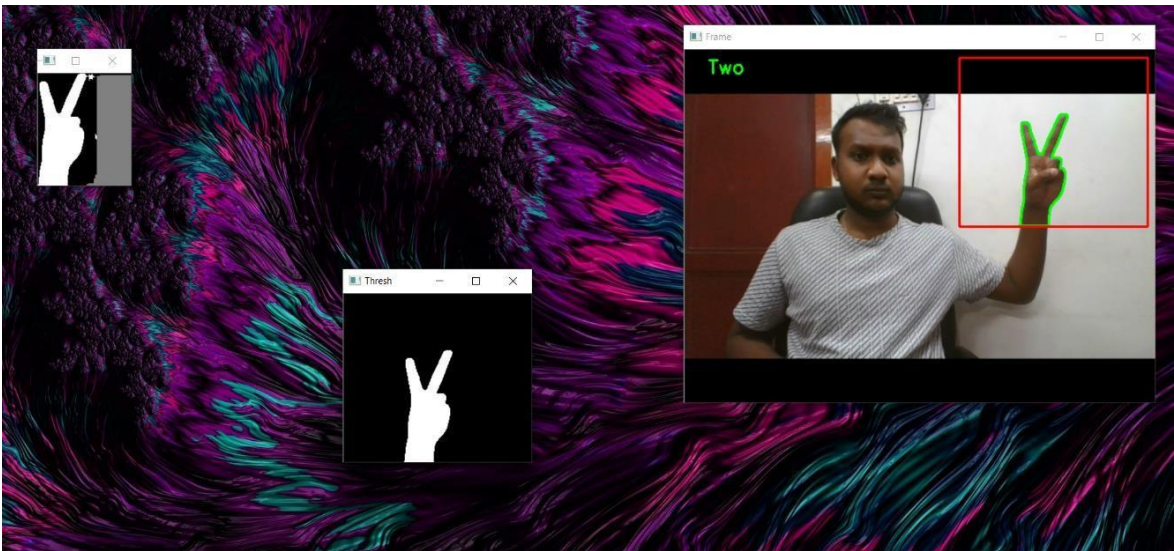
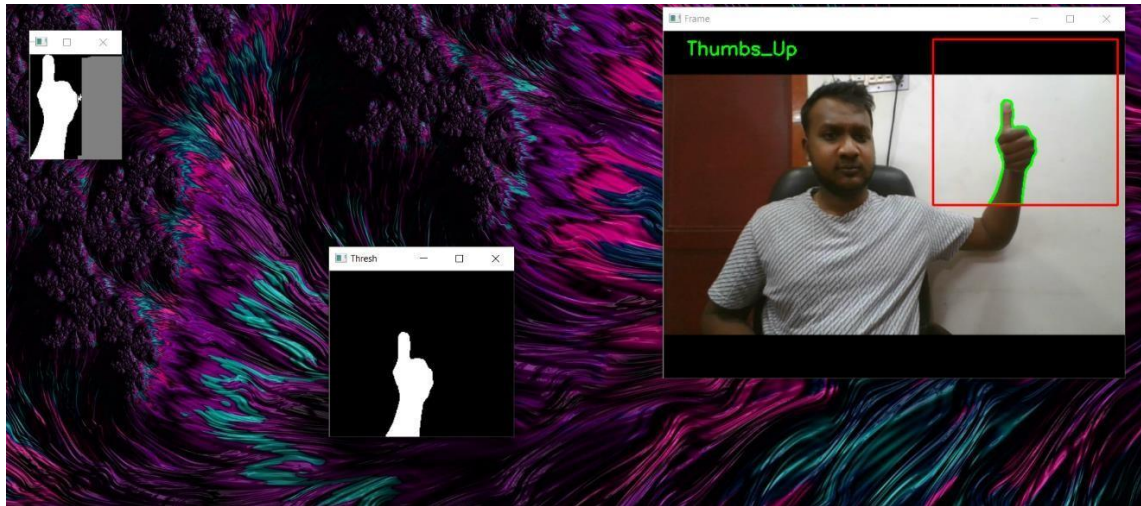


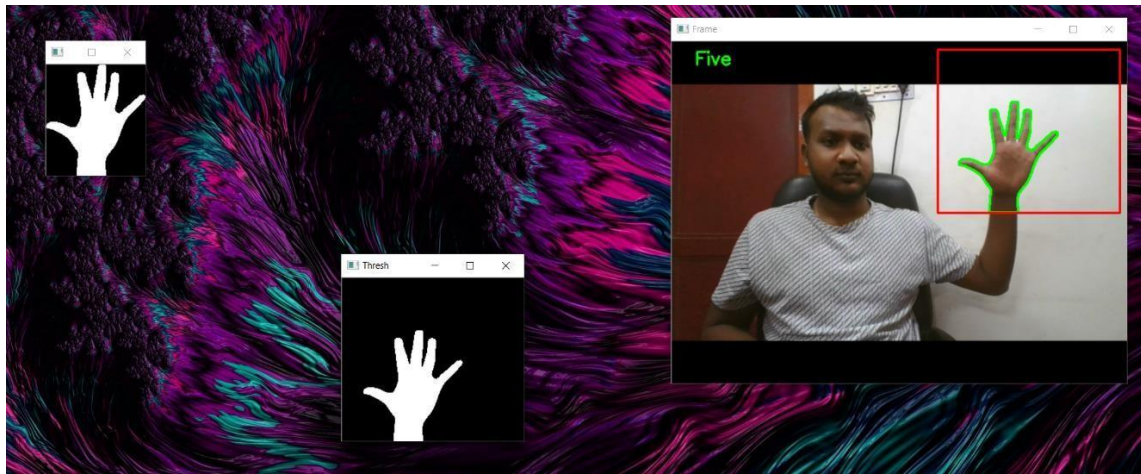












## C.PAPER PUBLICATION

# SIGN LANGUAGE DECODER FOR THE HEARING AND SPEAKING IMPAIRED

JOSHUA KENNETH PAUL J, HARIHARAN A, JANY SHABU S.L. M.TECH, (PHD)  
Department of CSE, Sathyabama Institute of Science and Technology

### ABSTRACT:

Sign language, the sole medium of contact for dumb and deaf people. These visually disabled people communicate their feelings and ideas to other persons with the aid of sign language. As it is difficult for the normal human to understand these languages, these mentally disabled individuals always have to hold the interpreter along for connecting with the world. Recognition of sign's has since become an analytical assignment. Since sign language contains a variety of motions and hand gestures, the consistency of sign's relies on the exact recognition of hand gestures. This paper presents a novel approach for sign language recognition. The primary aim is to correctly classify and identify the hand gesture. This work presents skin color-based segmentation models for hand region and detection from background, HOG method is used, extracts the shape information features and finally Random Forest machine learning classifier is utilized with effective results. The accuracy of our new models was revealed to be considerably higher than existing one.

**Keywords:** Sign Language, Hand Gesture, Skin Segmentation, Histogram Of Gradients (HOG), Machine Learning, Random Forest, Indian Sign Language

### I. INTRODUCTION

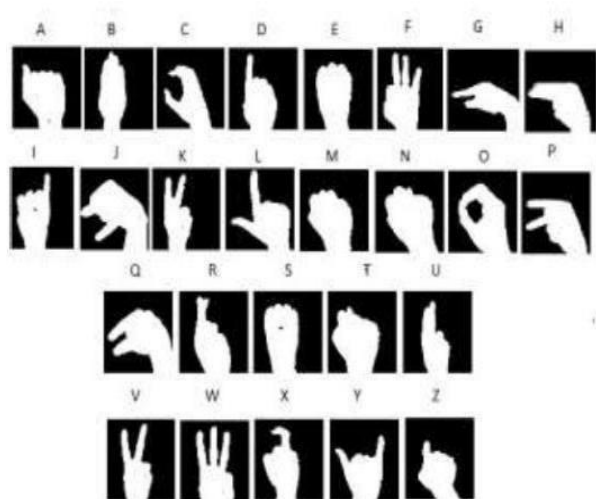
Sign language, the commonly used means for communicating by people who are deaf-dumb. There's nothing but, sign language which consists of various signs shaped by different hand forms, their motions, orientations, and many facial expressions.

There are almost 466.5 million people who suffer hearing loss globally, and 34.5 million of them are youth. Deaf people have no or very little hearing capacity. For communicating, they use sign language. Different places of the world, people use numerous sign languages.

India has its own sign language, Indian Sign Language, under the name (ISL). A few schools for deaf children remain in developed nations. In developed countries, unemployment rates are high for people having hearing loss.

According to data from Ethnologue, the rate of students attending school's literacy rate are much lower among the deaf community in India, that is around 1 percent of the total population. It goes on that official acceptance of sign languages dramatically boosts usability, enhances the availability of interpreters, offers documentation of sign languages. Finger spelling, the illustration of writing system's letters and occasionally numeral systems. Using finger spelling, Sign Language will represent the English alphabet A-Z. Maybe one-handed or two-handed, and the two-handed form is followed by ISL. Several scholars have suggested various forms of hand gesture edge detection techniques in the last year, but these techniques have demonstrated several disadvantages, such as false edge detection, broken edges, high calculation time, etc. In edge configuration, edge orientation and noise setting, the set of edge detection variables is constrained.

This paper approaches identifying sign language and to recognize it by image processing segmentation through feature extraction techniques and the machine learning model. It's been implemented by skin color segmentation; HOG feature extraction; Random Forest classification models.



**Fig.1 Samples of Indian Sign Language**

## II. PREVIOUS WORKS

Previously, less work was performed on the ISL. In [5], the main techniques involve key point image recognition using SIFT and then comparing new image key point with the regular image key point for every alphabet in the database to identify new image with the mark of the nearest matches.

The eigen vectors of covariance matrix is calculated from image representation vector and used the Euclidean distance of new image proprietary vector from those in the newest image classification training dataset [1].

In [4], Using Neural networks in preparation, but the dataset consists only of single-handed images and extracted feature vectors is centered on angle between fingertips, number of fingers.

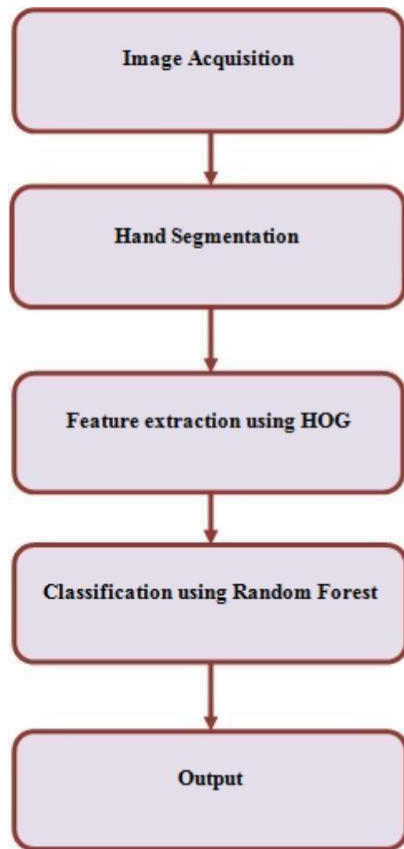
With all these works, the major issue was about datasets, it was never mentioned anywhere about where did the datasets come and from whom were it taken. Authors may have formed on getting sets from the people who works for the thesis.

It's possible the authors compiled the dataset from single source of team members in a thesis for related work we found at [3] and split testing and training sets before revealing the accuracy. Only a single subset of alphabets was tested.

## III. PROPOSED METHODOLOGY

The proposed sign language recognition done as per below steps:

1. The first step, segment the portion of the skin (hand region) from the entire image (background), as remaining part of character classification can be considered as noise.
2. The second step, derive relevant features or characteristics from hand segmented images that can prove useful, i.e., classification, for the next stage.
3. As described above, the third step, input the derived features to supervised learning model for training process and use the learned models for classification.



**Fig.2 Overall Proposed Model**

## A. IMAGE ACQUISITION

In sign recognition, the foremost step is, Interfacing with the camera is very important aspect. To catch hand gesture, a webcam is used. The webcam included in laptops or an external camera is used to interface. But it's important to mandatory be in high res for captured images. So, selecting a decent webcam & its interface is an essential task of this technique.

## B. HAND SEGMENTATION

Image preprocessing involves cropping, filtering, change of brightness, contrast and others. Image segmentation & image cropping techniques used to do some process enhancement. There are captured images form of RGB. The foremost step, transform images in RGB to Binary, so that unused portions of images can be eliminated by cropping images.

In image segmentation, skin colour segmentation model is proposed which used to segment the hand from background and used for next feature extraction process.



**Fig.3 Hand (Skin) Segmentation Results**

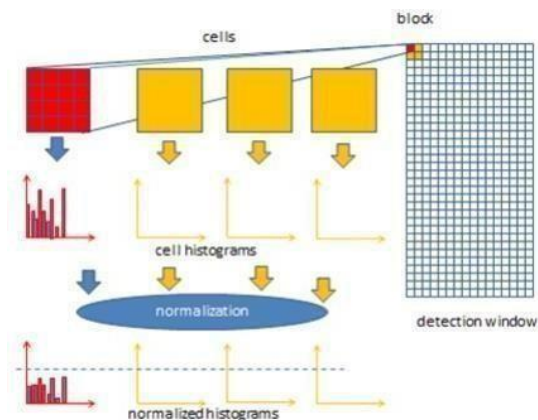
## C. FEATURE EXTRACTION:

Extracting features is very helpful step in building a sign recognition database. For letter image similarity characterization, both global and local visual features procured define the different visual concepts of alphabets in the manual efficiently. In sign recognition, there are primarily two kinds that makes feature extraction technique involved. First is shape representation and description approaches based on Contour, and second is shape representation and description techniques based on Region. Among these chosen are those focused on implementation processes. Here we propose HOG feature descriptor method for extracting feature which is applied handsegmented image. HOG method is presented to extracting shape features.

### Histogram of Gradients (HOG)

The descriptors of HOG features (shown in Fig.4) are always similar to descriptors of SIFT features, but different in the way they are calculated on grid of differently yet uniformly spaced cells and uses overlapping for local contrast normalization to enhance precision.

Images were analyzed using the HOG approach and features were used efficiently. Since HOG is a descriptor of the feature, its key duty is to take the image and extract the important component from image and then reject all the remaining redundant parts. The size of the file, which is the width and height of the image, along with the 3 channels, is taken into consideration in determining the attribute descriptors. An attribute vector is computed using all this information and then information is passed and on into the classifier to be remembered. First, the horizontal and vertical gradients should be measured to determine the HOG function descriptor.



**Fig.4 HOG Features**

It is also possible to take several steps to optimize the HOG solution, such as gamma correction. The image broke as many cells in which there are 2 values assigned to each cell pixel, which are direction and magnitude, helping to compactly represent the image.

## D. CLASSIFICATION

Above-generated feature vector is for each image is trained through by classifiers like Random Forest, and finally that will be the last technique. Accuracy is calculated for model and by passing them after the testing set.

### Random Forest

Random Forest, the popular most model in supervised machine learning models. RF, an ensemble model which works for regression and classification. So, it is called as ensemble random forest that is a model of integrating several classifiers ordered to solve complex problems and enhance efficiency of the model.

In RF, before having result multiple decision trees used. Thus, combining many trees for decision. The working can be illustrated in the below steps and diagram:

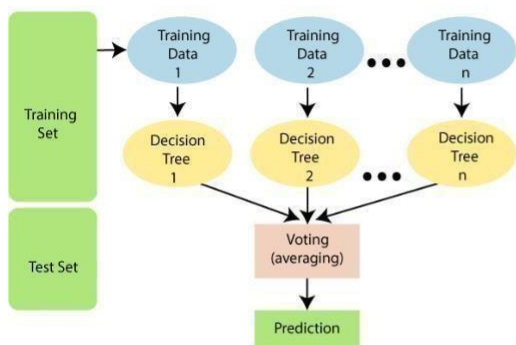
Step 1: From trainingset, pick random P data points.

Step 2: Form decision trees correlated with the chosen data points (subsets).

Step 3: Denote the number of decision trees to build as N.

Step 4: Repeat initial two steps.

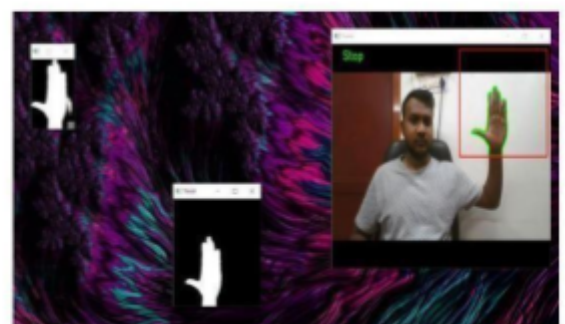
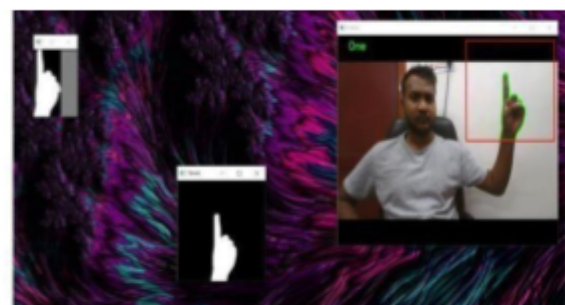
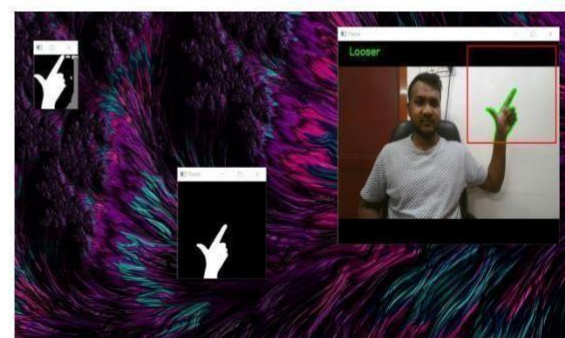
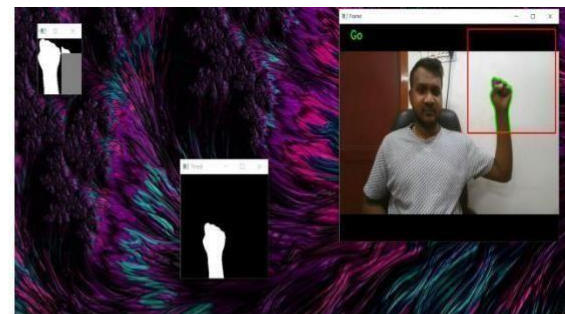
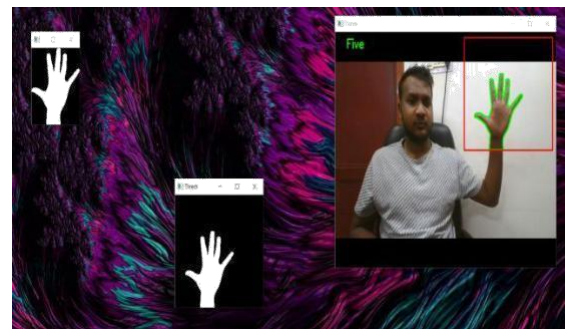
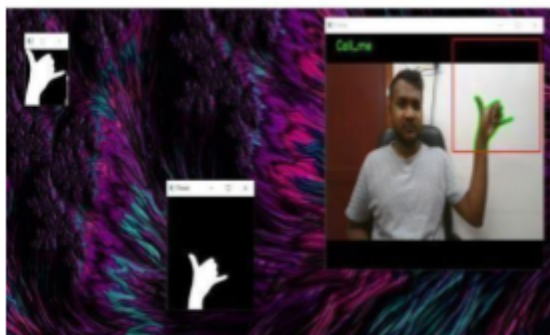
Step 5: Appraise the result of every decision tree with fresh data points and allocate them to a group based on majority votes.

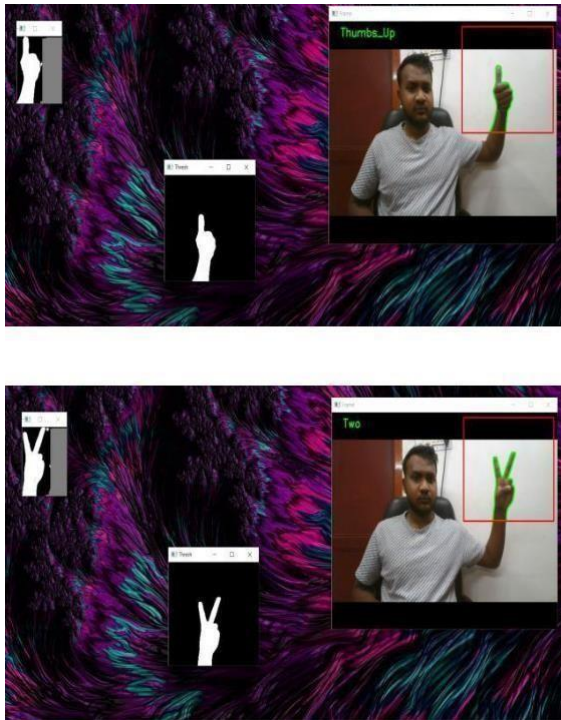


**Fig. 5 Working of Random Forest**

## IV. EXPERIMENTAL RESULTS

The system uses the approach of machine learning classification to identify hand gestures such as alphabets or numbers or words so that the system can predict the precise outcome of the hand sign and make the words as phrases.





**Fig.6 Sign Recognition Results by Proposed Model**

## V. CONCLUSION

In this paper, we used novel methodology to present automated hand sign language gesture recognition in real time. Signlanguage recognition system was successfully implemented by hand detection using skin color segmentation, shape feature extraction using HOG and classification using machine learning model i.e., Random Forest. The proposed model obtains the highest accuracy like **94.3%** by HOG and Random Forest classifiers mixed. This work will be extended in the future to enhance obtained results and create a Sign-Voice application that helps people with listening impairments to converse more fluently with people.

## REFERENCES

- [1] JOYEETA SINGH, K. D. "Indian sign language recognition using eigen value weighted euclidean distance-based classification technique", International Journal of Advanced Computer Science and Applications, 2013.

- [2] NEHA V. TAVARI, P. A. V. Debrancher. "Indian sign language recognition based on histograms of oriented gradient", International Journal of Computer Science and Information Technologies Vol 5 (3), 2014.
- [3] NMANIVAS. Gesture recognition system. <https://github.com/nmanivas/GestureRecognitionSystem>.
- [4] PADMAVATHI . S, SAIPREETHY.M.S, V. "Indian sign language character recognition using neural networks", IJCA Special Issue on Recent Trends in Pattern Recognition and Image Analysis, RTPRIA, 2013.
- [5] SAKSHI GOYAL, ISHITA SHARMA, S. S. "Sign language recognition system for deaf and dumb people", International Journal of Engineering Research Technology, April 2013.
- [6] Yiwen He, Jianyu Yang, Zhanpeng Shao, Youfu Li, "Salient feature point selection for real time RGB- D hand gesture recognition", IEEE International Conference on real- time Computing and Robotics, 2017.
- [7] Rania A. Elsayed, Mahmoud I. Abdalla, Mohammed S. Sayed, "Hand gesture recognition based on dimensionality reduction of histogram of oriented gradients", Japan-Africa Conference on Electronics, Communication and Computers, 2017.
- [8] Himadri NathSaha, SayanTapadar, Shinjini Ray, Suhrid Krishna Chatterjee, "A Machine Learning based approach for Hand Gesture Recognition using distinctive feature extraction",IEEE 8th Annual Conference on Computing and Communication, 2018
- [9] Mukesh Kumar Makwana, " Sign Language Recognition", Indian Institute of Science, Bangalore.
- [10] Escalera, Sergio, et al. "Chalearn looking at people challenge 2014: Dataset and results", Workshop at the European Conference on Computer Vision, Springer International Publishing, 2014.

[11] R. Saxena, S. Hussain, X. Han, H. Shin and J. A. Khan, "Hand gesture recognition using deep learning", International SoC Design Conference (ISOCC) Seoul, 2017.

[12] Takayoshi Yamashita and Taro Watasue, "Hand posture recognition based on bottom-up structured deep convolutional neural network with curriculum learning", IEEE International Conference on Image Processing (ICIP) Paris, 2014.

## D. PLAGIARISM REPORT

# RP013

*by* S Report

---

**Submission date:** 17-Mar-2021 08:17PM (UTC-0700)

**Submission ID:**

1534390118 **File name:**

hari.pdf (518.31K) **Word**

**count:** 1868

**Character count:** 10609

# SIGN LANGUAGE DECODER FOR THE HEARING AND SPEAKING IMPAIRED

JOSHUA KENNETH PAUL J, HARIHARAN A, JANY SHABU SL. M.TECH,  
(PHD)

**ABSTRACT:** Department of CSE, Sathyabama Institute of Science and Technology

Sign language, the sole medium of contact for dumb and deaf people. These visually disabled people communicate their feelings and ideas to other persons with the aid of sign language. As it is difficult for the normal human to understand these languages, these mentally disabled individuals always have to hold the interpreter along for connecting with the world. Recognition of sign language has since become an analytical assignment. Since sign language contains a variety of motions and hand gestures, the consistency of sign language relies on the exact recognition of hand gestures. This paper presents a novel approach for sign language recognition. The primary aim is to correctly classify and identify the hand gesture. This work presents skin color-based segmentation models for hand region and detection from background. HOG method is used to extract the shape information features and finally Random Forest machine learning classifier is utilized with effective results. The accuracy of our new models was revealed to be considerably higher than existing ones.

**Keywords:** Sign Language, Hand Gesture, Skin Segmentation, Histogram of Gradients (HOG), Machine Learning, Random Forest, Indian Sign Language

## I. INTRODUCTION

Sign language, the commonly used means for communicating by people who are deaf-dumb. There's nothing but, sign language which consists of various signs shaped by different hand forms, their motions, orientations, and many facial expressions.

There are almost 466.5 million people who suffer hearing loss globally, and 34 million of them are youth. Deaf people have no or very little hearing capacity. For communicating, they use sign language. Different places of the world, people use numerous sign languages.

India has its own sign language, Indian Sign Language, under the name (ISL). A few schools for deaf children remain in developed nations. In developed countries, unemployment rates are high for people having hearing loss.

According to data from Ethnologue, the rate of students attending school's literacy rate are much lower among the deaf community in India, that is around 1 percent of the total population. It goes on that official acceptance of sign languages dramatically boosts usability, enhances the availability of interpreters, offers documentation of sign languages. Finger spelling, the illustration of writing system's letters and occasionally numeral systems. Using finger spelling, Sign Language will represent the English alphabet A-Z. Maybe one-handed or two-handed, and the two-handed form is followed by ISL. Several scholars have suggested various forms of hand gesture edge detection techniques in the last year, but these techniques have demonstrated several disadvantages, such as false edge detection, broken edges, high calculation time, etc. In edge configuration, edge orientation and noise setting, the set of edge detection variables is constrained.

This paper approaches identifying sign language and to recognize it by image processing segmentation through feature extraction techniques and the machine learning model. It's been implemented by skin color segmentation; HOG feature extraction; Random Forest classification



Fig.1 Samples of Indian Sign Language

## II. PREVIOUS WORKS

Previously, less work was performed on the ISL. In [5], the main techniques involve key point image recognition using SIFT and then comparing new image key point with the regular image key point in every alphabet in the database to identify new image with the mark of the nearest matches.

The eigen vectors of covariance matrix is calculated from image representation vector and using the Euclidean distance of new image proprietary vector from those in the newest image classification training dataset [1].

In [4], using Neural network preparation, but the dataset consists only of single-handed images and extracted feature vectors is centered on angle between fingertips, number of fingers.

With all these works, the major issue was about datasets, it was never mentioned anywhere about where did the datasets come and from whom were it taken. Authors may have focused on getting sets from the people who works for the thesis.

It's possible the authors compiled dataset from single source of team members in a thesis for related work we found at [3] and split testing and training sets before revealing the security. Only a single subset of alphabets was tested.

## III. PROPOSED METHODOLOGY

The proposed sign language recognition done

1. The first step, segment the portion of the skin (hand region) from the entire image as remaining of (background), noise. part character classification can be considered as
  2. The second step, derive relevant features or characteristics from hand segmented images that can prove useful, i.e., classification, for the next stage.
- As described above, in the first step, input the derived features to supervised learning model for training process and use the learned models for classification.



**Fig.2 Overall Proposed Model**

## A, IMAGE ACQUISITION

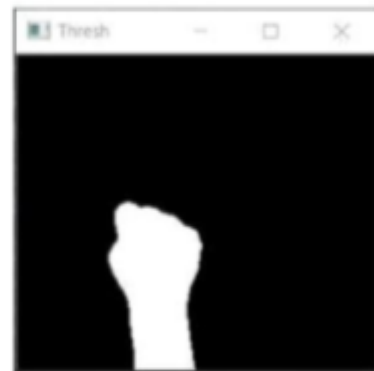
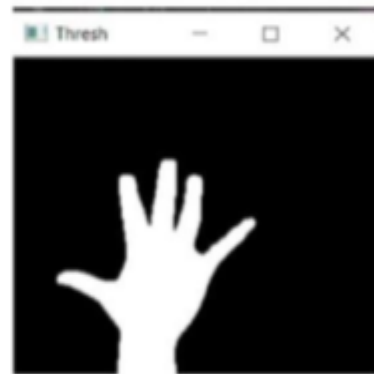
In sign recognition, the foremost step is, Interfacing with the camera is very important aspect. To catch hand gesture, a webcam is used. The webcam included in laptops or an external camera is used to interface. But it's important to mandatory be

high res or captured using decent webcam bits interface is an essential task of

## B. HAND SEGMENTATION

Image preproessing involves cropping, filtering, change of brightness, contrast and others Image segmentation & image cropping techniques used to do some processenhancement. There are captured images form of RGB. The foremost step, transform images in RCiB to Binary, so that unused portions of imagescan be eliminated by cropping

In image segmentation, skin color segmentation model is proposed which used to segment the hand from background and used for next tea excretion process.



**Fig.3 Hand (Skin) Segmentation Results**

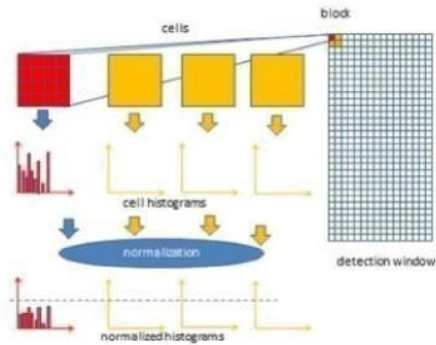
### C. FEATURE EXTRACTION:

Extracting features is very helpful step in building a sign recognition database. For letter image similarity characterization, both global and local visual features procured define the different visual concepts of alphabets in the manual efficiently. In sign recognition, there are primarily two kinds that makes feature extraction technique involved. First is shape representation and description approaches based on Contour, and second is shape representation and description techniques based on Region. Among these chosen are those focused on implementation processes. Here we propose HOG feature descriptor method for extracting feature which is applied hand segmented image. HOG method is presented to extracting shape features.

#### Histogram of Gradients (HOG)

The descriptors of HOG features (shown in Fig.4) are always similar to descriptors of SIFT features, but different in the way they are calculated on grid of differently yet uniformly spaced cells and uses overlapping for local contrast normalization to enhance precision.

Images were analyzed using the HOG approach and features were used efficiently. Since HOG is a descriptor of the feature, its key duty is to take the image and extract the important component from image and then reject all the remaining redundant parts. The size of the file, which is the width and height of the image, along with the 3 channels, is taken into consideration in determining the attribute descriptors. An attribute vector is computed using all this information and then information is passed and on into the classifier to be remembered. First, the horizontal and vertical gradients should be measured to determine the HOG function descriptor.



It is also possible to take several steps to optimize the HOG solution, such as gamma correction. The image broke as many cells in which there are 2 values assigned to each cell pixel, which are direction and magnitude, helping to compactly represent the image.

### D. CLASSIFICATION

After the feature vector is for each image is trained through by classifiers like Random Forest, and finally that will be the last technique. Accuracy is calculated for model and by passing them after the testing set.

#### Random Forest

Random Forest, the popular most model in supervised machine learning models. RF, an ensemble model which works for regression and classification. So, it is called as ensemble random forest that is a model of integrating several classifiers ordered to solve complex problems and enhance efficiency of the model.

In RF, before having result multiple decision trees used. Thus, combining many trees for decision. The working can be illustrated in the below steps and diagram:

Step 1: From training set, pick random P data points.

Step 2: Form decision trees correlated with the chosen data points (subnets).

Step 3: Note the number of decision trees to build as N.

Repeat initial two steps.

Step 5: Appraise the result of every decision tree with fresh data points and allocate them to a group based on majority votes.

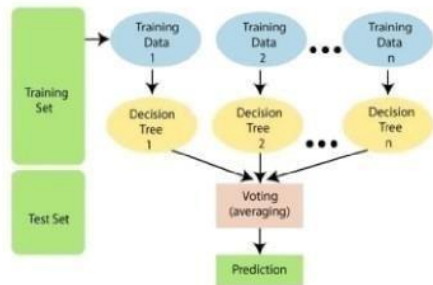
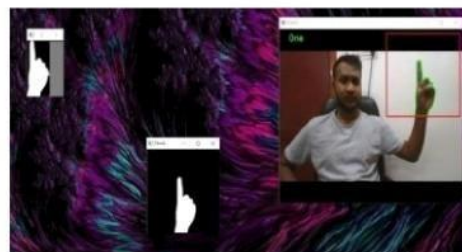
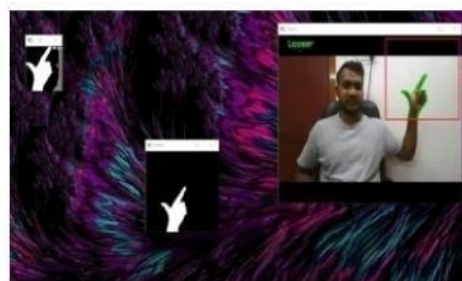
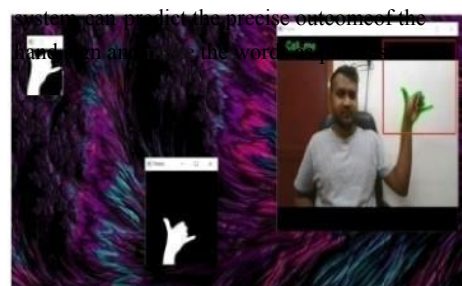


Fig . 5 WOPkIng Of RBHdOm FOPflGt

## IV. EXPERIMENTAL RESULTS

The system uses the approach of machine learning classification to identify hand gestures such as alphabets or numbers



## Fig. Sign Recognition Remits try Proposed Model

## V. CONCLUSION

In this paper, we used novel methodology to present automated hand sign language gesture recognition in real time. Sign language recognition system was successfully implemented by hand detection using skin color segmentation, shape feature extraction using HOG and classification using machine learning model i.e., Random Forest. The proposed model obtains the highest accuracy like 94.7% by HOG and Random Forest classifiers mixed. This work will be extended in the future to enhance obtained results and create a Sign-Voice application that helps people with listening impairments to converse more fluently with people.

## REFERENCES

[1] JOYEETA SINGH, K. D. "Indian sign language recognition using eigen value weighted euclidean distance-based classification technique", International Journal of Advanced Computer Science and Applications, 2013.

[2] NEHA V. TAVARI, P. A. V. Debrancher. "Indian sign language recognition based on histograms of oriented gradient", International Journal of Computer Science and Information Technologies Vol 5 (3), 2014.

[3] NMANIVAS. Gesture recognition system.  
<https://github.com/nmanivafGestureRecognitionSyst>

[4] PADMAVATHI S, SAIPREETHY M.S, V. "Indian sign language character recognition using neural networks", UCA Special Issue on Recent Trends in Pattern Recognition and Image Analysis, RTPRIA, 2013.

[5] SAKSHI GQYAL, ISHITA SHARMA, S. S. "Sign language recognition system for deaf and dumb people", International Journal of Engineering Research Technology, April 2013.

[6] Yiwen He, Jianyu Yang, Zhanpeng Shao, Youfu Li, "Salient feature point selection for real time RGB-D hand gesture recognition", IEEE International Conference on real-time Computing and Robotics, 2017.

[7] Rania A. Elsayed, Mahmoud I. Abdalla, Mohammed S. Sayed, "Hand gesture recognition based on dimensionality reduction of histogram of oriented gradients", Japan-Africa Conference on Electronics, Communication and Computers, 2017.

[8] Himadri Nath Saha, Sayan Tapadar, Shinjini Ray, Suhrid Krishna Chatterjee, "A Machine Learning based approach for Hand Gesture Recognition using distinctive feature extraction", IEEE 8th Annual Conference on Computing and Communication, 2018

[9] Mukesh Kumar Makwana, "Sign Language Recognition", Indian Institute of Science, Bangalore.

[10] Escalera, Sergio, et al. "Otalearn looking at people challenge 2014: Dataset and results", Workshop at

the European Conference on Computer  
Vision, Springer International Publishing,  
2014.

[11] R. Saxena, S. Hussain, X. Han, H. Shin and J. A. Khan, "Hand gesture recognition using deep learning", International SoC Design Conference (ISOCC) Scout, 2017.

[12] Takayoshi Yamarhita and Taro Watasue, "Hand posture recognition based on bottom-up structured deepconvolutional neural network with curriculum learning", IEEE International Conference on Image Processing (IC&) Paris, 2014.

ORIGINALITY REPORT



SIMILARITY INDEX      INTERNET SOURCES      PUBLICATIONS      STUDENT PAPERS

PRIMARY SOURCES

**1** [cse.iitk.ac.in](http://cse.iitk.ac.in) 4%  
Internet Source

Submitted to Malla Reddy Engineering College, Hyderabad 4%  
Student Paper

**3** Submitted to The Academy of Korean Studies  
Student Paper

**4** [etheses.whiterose.ac.uk](http://etheses.whiterose.ac.uk)  
Internet Source

**5** quan Yang. "Chinese sign language recognition based on gray-level co-occurrence

matrix and other  
multi-features fusion",  
2009 4th IEEE  
Conference on Industrial  
Electronics and  
Applications, 05/2009

Publication

1%

---

1%

---

1%



"Computational  
Vision and  
Bio-Inspired  
Computing",  
Springer Science  
and Business Media  
LLC, 2020

Publication

<1%

---

---

|                |     |                 |     |
|----------------|-----|-----------------|-----|
| Exclude quotes | Off | Exclude matches | Off |
| bibliography   | Off |                 |     |