**A PROJECT REPORT**

**ON**

**HAND GESTURE RECOGNITION SYSTEM**

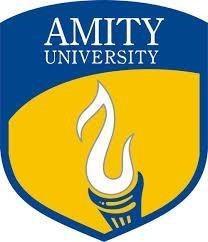
***In partial fulfilment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

***In***

**ELECTRONICS & COMMUNICATION ENGINEERING**

******

**Guide: Submitted by:**

Mr. Rajkumar Gupta Krishna Sidhvi

**DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING**

**AMITY UNIVERSITY RAJASTHAN**

**May 2021**

**DECLARATION**

I hereby declare that this submission is my own work and to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

.

**N.** Krishna Sidhvi(A20405117004)

Date:06/2021

Counter signed by

**Mr. Rajkumar Gupta.**

### CERTIFICATE

I hereby certify that the work which is being presented in the B.Tech. Major Project Report entitled **“Hand Gesture Recognition System”,** in partial fulfilment of the requirements for the award of the **Bachelor of Technology in ELECTRONICS & COMMUNICATION ENGINEERING** and submitted to the Department of Electronics & Communication engineering of *Amity School of Engineering and Technology, Amity University Rajasthan* is an authentic record of my own work carried out during a period from January 2018 to May 2018 under the supervision of **Mr. Rajkumar Gupta, ECE** **Department**.

The matter presented in this thesis has not been submitted by me for the award of any other degree elsewhere.

N. Krishna Sidhvi

Date: 06/2021

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

***Mr. Rajkumar Gupta***

06/2021

**ACKNOWLEDGEMENT**

I owe special gratitude to my parents for their persistent efforts in me.

I would also like to extend a special thanks to all my faculties for their support and confidence in me while carrying out the study.

This report would not have been possible without the able guidance and support of my supervisor and mentor Mr. Rajkumar Gupta.

My thanks go to all the supporters from the Department of Electronics and Communication who helped me to complete this report. Their willingness to share their experience provided valuable inputs and comments which have helped me to charter a course of action in improving this report.

N. Krishna Sidhvi

**ABSTRACT**

Inability to speak is considered to be true disability. People with this disability use different modes to communicate with others. There are number of methods available for their communication one such common method of communication is sign language. Developing sign language application for deaf and dumb people can be very important, as they’ll be able to communicate easily with even those who can’t understand sign language. Our project aims at taking the basic step in bridging the communication gap between normal people, deaf and dumb people using sign language. The main focus of this work is to create a vision-based system to identify sign language gestures from the video sequences. The reason for choosing a system based on vision relates to the fact that it provides a simpler and more intuitive way of communication between a human and a computer.

### TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **1** | **1.1** INTRODUCTION | **1** |
| **1.2** RELEVANCE OF THE PROJECT | **2** |
| **1.3** PROBLEM STATEMENT | **3** |
| **1.4** EXISTING SYSTEM | **3** |
| **1.5** LIMITATION OF EXISTING SYSTEM | **4** |
| **1.6** PROPOSES SYSTEM | **4** |
| **1.7** ADVANTAGES OF PROPOSED SYSTEM | **5** |
| **1.8** AIM AND OBJECTIVE | **5** |
| **2** | LITERATURE SURVEY | **6** |
| **3** | **3.1** SYSTEM REQUIREMENTS | **14** |
| **3.1.1** Feasibility Study | **14** |
| **4** | **4.1** SYSTEM REQUIREMENTS SPECIFICATION | **17** |
| **4.1.1** Requirement specification | **17** |
| **4.1.2** Software Description | **18** |
| **4.1.3** Modules | **21** |

|  |  |  |
| --- | --- | --- |
|  | **4.2** SYSTEM DESIGN | **22** |
| **4.2.1** System Architecture | **22** |
|  | **4.2.3** Data Flow Diagram | **23** |
|  | **4.2.2** Sequence Diagram | **25** |
|  | **4.2.4** Use case Diagram | **26** |
|  | **4.2.5** Flow chart | **28** |
| **5** | **5.1** IMPLEMENTATION | **29** |
|  | **5.1.1** Pre-Processing | **29** |
| **5.1.2** Feature Extraction of Processed Image. | **31** |
|  | **5.1.3** Classification | **31** |
|  | **5.2** CODING | **32** |
| **6** | **6.1** TESTING AND TYPES | **42** |
|  | **6.1.1** Testing | **42** |
|  | **6.2** TYPES OF TESTING | **44** |
|  | **6.2.1** Unit Testing | **44** |
|  | **6.2.2** Integration Testing | **45** |
|  | **6.2.3** System Testing | **45** |
|  | **6.2.4** Performance Testing | **45** |
|  | **6.2.5** Validation Testing | **45** |
|  | **6.2.6** Black Box testing | **46** |
|  | **6.2.7** White Box Testing | **46** |

|  |  |  |
| --- | --- | --- |
|  | **6.2.8** Acceptance Testing | **46** |
| **7** | **7.1** OUTPUT | **49** |
|  | **7.2** CONCLUSION | **51** |
|  | **7.3** FUTURE WORKS | **51** |
|  | REFERENCES | **52** |

### LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Fig 1** | Overview of the proposed system in five modules. | **23** |
| **Fig 2** | Overview of the High-Level Block Diagram. | **24** |
| **Fig 3** | Data Flow Diagram. | **25** |
| **Fig 4** | Overview of Sequence Diagram. | **26** |
| **Fig 5** | Overview of Use Case Diagram. | **27** |
| **Fig 6** | Flow Chart. | **28** |
| **Fig 7** | System Implementation. | **29** |
| **Fig 8** | Input Image and Output screen 1 | **50** |
| **Fig. 9** | Output Screen and Input image 2 | **50** |
| **Fig. 10** | Input Image and Output screen 3 | **51** |
| **Fig 11** | Output Screen and input Image 4 | **51** |

#### Chapter 1

##### Introduction

Image processing is a rapidly growing area in diverse applications, such as multimedia computing, secured data communication, biomedical, biometrics, remote sensing, texture understanding, pattern recognition, content-based retrieval, compression, and many more. This is all about how a computer can sense pictorial data after processing an image. Among the set of gestures intuitively performed by humans when communicating with each other, pointing gestures are especially interesting for communication and is perhaps the most intuitive interface for selection. They open up the possibility of intuitively indicating objects and locations, e.g., to make a robot change moving direction or simply mark some object. This is particularly useful in combination with speech recognition as pointing gestures can be used to specify parameters of location in verbal statements. This technology can be a boon for disable people who are not able to speak hence can’t communicate. Also, if the person has different language than receiver, then also, it can be used to as translator. There has been always considered a challenge the development of a natural interaction interface, where people interact with technology as they are used to interact with the real world. A hand free interface, based only on human gestures, where no devices are attached to the user, will naturally immerse the user from the real world to the virtual environment.

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

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

##### Relevance of the project

Sign Hand Gestures are used in day-to-day life like nodding and waving without us being aware of them. It has become an important part in the communication among the humans. In the recent years new methods of Human Computer Interaction (HCI) are being developed. Indian Sign Language is based on hand sign.

##### Problem Statement

The main problem in the existing system is that user need to be in front of the laptop to use it. If they are busy with some other work and need to any application in the laptop, they need to go in front of the laptop and do. It is well known fact that communication is very essential in day-to-day life. Humans with disabilities often experience issues in this area. [Deaf and Dumb]. In the world of sign language, and gestures, a lot of research work has been done over past few years. Sign Language Recognition is one of the most growing fields of research area. Language is mainly used for communication of deaf-dumb people. In present scenario, it is impossible for humans to understand the sign language without any practice. We hereby facilitate a human machine interactive system that would be very helpful for communication between deaf and dumb people and humans in real world situation.

##### Existing System

Existing proposed a technique of first computing the similarity of different gestures and then assign probabilities to them using Bayesian Interface Rule. These classes consist of Hu moments with geometrical attributes like rotation, transformation and scale in variation which were used as features for classification. Performance of this technique was very well and it was giving 95 °/o accurate results. Existing system also proposed a similar technique which also uses H U-moments along with modified KNN (K-Nearest Neighbor) algorithm for classification called as Locally Weighted Naive Bayes Classifier.

Classification results where this technique was 93°/o accurate under different lighting conditions with different users.

##### Limitation of Existing System

* Irrelevant object might overlap with the hand. Wrong object extraction appeared if the objects larger than the hand.
* Performance recognition algorithm decreases when the distance is greater than 1.5 meters between the user and the camera.
* System limitations restrict the applications such as the arm must be vertical, the palm is facing the camera and the finger color must be basic color such as either red or green or blue.
* Ambient light affects the color detection threshold.

##### Proposed System

The first step for our proposed system is the capturing of the video using webcam where different alphabets were taken into consideration. Skin Filtering was performed to the input video frames for detection of hand gestures. It was done so that the required hand could be extracted from the background. Skin Filtering is a technique used for separating the skin-colored regions from the non-skin-colored regions. In our proposed system there are

5 modules: real time Input image from webcam, preprocessing and segmentation, feature extraction, classification and Results analysis (gesture recognition). For gesture recognition is real time recognized in live camera

. The proposed system is used in SVM (Support Vector Machine), K- Neighbors-Classifier, Logistic-Regression, MLP-Classifier, Naive Bayes, Random-Forest-Classifier algorithms.

This system can definitely help millions of deaf people to communicate with other normal people. A fast, novel and robust system was proposed for recognition of different alphabets of Indian Sign Language for video sequences. The proposed system is a real time video processing that is based on a real time application system.

##### Advantage

* There are no moving parts, so device wear is not an issue.
* We have proposed a system which is able to recognize the various alphabets of Indian Sign Language for Human-Computer interaction giving more accurate results at least possible time.
* Accuracy rate obtained was 98% but it lacks proper Skin filtering with changes in illumination.
* Proper classifier was used to recognize the gestures.
* SVM (Support Vector Machine), K-Neighbors-Classifier, Logistic- Regression, MLP-Classifier, Naive Bayes, Random-Forest-Classifier Algorithms is used and after the training then output is achieved, thus giving the proper recognized gesture.

##### Aim and Objective

The main aim of the project is to provide an easy platform for the communication.

The project work focuses on the hand gesture using image processing. The main objective of the proposed system is to work as a medium of communication among deaf and dumb people to convey the message with normal person. A person who can talk and hear properly (normal person) cannot communicate with deaf & dumb person unless he/she is familiar with sign language

#### Chapter 2

##### 2.1 LITERATURE SURVEY

1. **Title: Using Multiple Sensors for Mobile Sign Language Recognition**

***Author:*** *Helene Brashear, Thad Starner, Paul Lukowicz & Holger Junker*

The authors built a constrained, lab-based Sign Language recognition system with the goal of making it a mobile assistive technology. They examine using multiple sensors for disambiguation of noisy data to improve recognition accuracy. The experiment compares the results of training a small gesture vocabulary using noisy vision data, accelerometer data and both data sets combined. The authors chose to use a rule–based grammar for sentence structure in the training and testing process. Speech recognition often uses statistical grammars for increased accuracy. These grammars are built by tying together phonemes (the simplest unit of speech) and training on the transition between the phonemes. The sets are usually done with bigrams (two phonemes tied together) or trigrams (three phonemes). Training using bigrams or trigrams requires considerably more data because representations of each transition of each word are now needed. In our case, the bigrams and trigrams would be built by tying together gestures. The current data set is too small to effectively train using bigrams or trigrams, but we intend to continue collecting data with the goal of implementing these techniques.

**Advantage:**

* + Benefit of the proposed design is that the user can monitor the camera’s view via the head mounted display.
  + Provides accuracy.

**Disadvantage:**

* + Data set is too small to effectively train using bigrams or trigrams.
  + The current system has only been trained on a very small vocabulary.

##### Title: A Vision Based Dynamic Gesture Recognition of Indian Sign Language on Kinect based Depth Images

***Author:*** *Geetha M, Manjusha C, Unnikrishnan P and Harikrishnan R*

Indian Sign Language (ISL) is a visual-spatial language which provides linguistic information using hands, arms, facial expressions, and head/body postures. The proposed work aims at recognizing 3D dynamic signs corresponding to ISL words. With the advent of

3D sensors like Microsoft Kinect Cameras, 3D geometric processing of images has received much attention in recent researches. The authors have captured 3D dynamic gestures of ISL words using Kinect camera and has proposed a novel method for feature extraction of dynamic gestures of ISL words. While languages like the American Sign Language (ASL) are of huge popularity in the field of research and development, Indian Sign Language on the other hand has been standardized recently and hence its (ISLs) recognition is less explored. The method extracts features from the signs and converts it to the intended textual form. The proposed method integrates both local as well as global information of the dynamic sign. A new trajectory-based feature extraction method using the concept of Axis of Least Inertia (ALI) is proposed for global feature extraction. An Eigen distance-based method using the seven3D key points- (five corresponding to each finger tips, one corresponding to center of the palm and another corresponding to lower part of palm), extracted using Kinect is proposed for local feature extraction. Integrating 3D local feature has improved the performance of the system as shown in the result. Apart from serving as an aid to the disabled people, other applications of the system also include serving as a sign language tutor, interpreter and also be of use in electronic systems that take gesture input from the users.

**Advantage:**

* + Improve the accuracy of recognition.
  + The proposed method integrates both local as well as global information of the dynamic sign.
  + Can handle different types of words in a common vision-based platform.

**Disadvantage:**

* + These methods are not user friendly and are more expensive.

##### Title: A Color Hand Gesture Database for Evaluating and Improving Algorithms on Hand Gesture and Posture Recognition

***Author:*** *Farhad Dadgostar, Andre L. C. Barczak, Abdolhossein Sarrafzadeh*

With the increase of research activities in vision-based hand posture and gesture recognition, new methods and algorithms are being developed. Although less attention is being paid to developing a standard platform for this purpose. Developing a database of hand gesture images is a necessary first step for standardizing the research on hand gesture recognition. For this purpose, we have developed an image database of hand posture and gesture images. The database contains hand images in different lighting conditions and collected using a digital camera. Details of the automatic segmentation and clipping of the hands are also discussed in this paper.

**Advantage:**

* + Automatically vary the lighting fairly in all directions and even produce very complex patterns of lighting by introducing more than one source of light.
  + Enable researchers to add their own backgrounds to the image or to use it as an object with known boundaries.

**Disadvantage:**

* + Unless some special gadgets are used to control the lighting, it is very difficult to vary the positions of the light fairly along the three axes.

##### Title: Low-cost approach for Real Time Sign Language Recognition

***Author:*** *Matheesha Fernando, Janaka Wijayanayaka*

Sign Language is the language of people who suffer from speech and hearing defects. Still the rest of the world doesn’t have a clear idea of sign language. The communication between speech impaired people and other people is very inefficient. To overcome this problem technology can act as an intermediate flexible medium for speech impaired people to communicate amongst themselves and with other individuals as well as to enhance their level of learning / education. The suggested solutions in the literature for sign language recognition are very expensive for day-to-day use. Therefore, the main objective of this research is to find out a low-cost affordable method of sign language interpretation. This paper discusses the possible ways to deal with the sign language postures to identify he signs and convert them into text and speech using appearance-based approach with a low-cost web camera. Further this approach I’ll be very useful to the sign language learners to practice sign language. During the research available human computer interaction approaches in posture recognition were tested and evaluated. A series of image processing techniques with Hub-moment classification was identified as the best approach. The system is able to recognize selected Sign Language signs with the accuracy of 76% without a controlled background with small light adjustments.

**Advantage:**

* + Helps in identifying a low cost, affordable method that can facilitate hearing and speech impaired people to communicate with the world in more comfortable way where they can easily get what they need from the society and also can contribute to the well-being of the society.
  + Can be used as a learning tool for sign language where hearing and speech impaired people can practice sign language using the application.

**Disadvantage:**

* + This project only looks at the hand postures not on hand gestures.

##### Title: MILES: Multiple-Instance Learning via Embedded Instance Selection

***Author:*** *Yixin Chen, Jinbo Bi and James Z. Wang*

Multiple-instance problems arise from the situations where training class labels are attached to sets of samples (named bags), instead of individual samples within each bag (called instances). Most previous multiple-instance learning (MIL) algorithms are developed based on the assumption that a bag is positive if and only if at least one of its instances is positive. Although the assumption works well in a drug activity prediction problem, it is rather restrictive for other applications, especially those in the computer vision area. The authors proposed a learning method, MILES (Multiple-Instance Learning via Embedded instance Selection), which converts the multiple-instance learning problem to a standard supervised learning problem that does not impose the assumption relating instance labels to bag labels. MILES maps each bag into a feature space defined by the instances in the training bags via an instance similarity measure. This feature mapping often provides a large number of redundant or irrelevant features. Hence 1-norm SVM is applied to select important features as well as construct classifiers simultaneously.

**Advantage:**

* + **Broad adaptability:** It provides a learning framework that converts a multiple- instance problem to a supervised learning problem.
  + **Low complexity:** It is efficient in computational complexity, therefore, can potentially be tailored to tasks that have stringent time or resource limits.
  + **Prediction capability:** In some multiple-instance problems, classification of instances is at least as important as the classification of bags.

**Disadvantage:**

* + The performance of MILES depends on whether there are “useful” features among those defined by the instances in the training bags.
  + In some applications, for example 3D object recognition, geometric constraints on the image patches are extremely useful in reducing the search space andimproving the recognition accuracy. However, MILES is not designed to take advantage of this type of prior knowledge.
  + The feature vectors generated by the mapping are not sparse.

##### Title: RGB-H-CbCr Skin Color Model for Human Face Detection

***Author:*** *Nusirwan Anwar bin Abdul Rahman, Kit Chong Wei and John See*

While the RGB, HSV and YUV (YCbCr) are standard models used in various color imaging applications, not all of their information is necessary to classify skin color. This paper

presents a novel skin color model, RGB-H-CbCr for the detection of human faces. Skin regions are extracted using a set of founding rules based on the skin color distribution obtained from a training set. The segmented face regions are further classified using a parallel combination of simple morphological operations. This model utilizes the additional hue and chrominance information of the image on top of standard RGB properties to improve the discriminality between skin pixels and non-skin pixels. In the proposed approach, skin regions are classified using the RGB boundary rules introduced by Peer et al. and also additional new rules for the H and CbCr subspaces. These rules are constructed based on the skin colour distribution obtained from the training images. The classification of the extracted regions is further refined using a parallel combination of morphological operations.

**Advantage:**

* + Able to deal with various brightness and illumination conditions as well as very effective compare to the other existing systems.

**Disadvantage:**

* + Doesn't provide success of a robust face detector.

##### Title: Robust Real-Time Face Detection

***Author:*** *Paul Viola and Michael J. Jones*

This paper describes a face detection framework that is capable of processing images extremely rapidly while achieving high detection rates. There are three key contributions. The first is the introduction of a new image representation called the “Integral Image” which allows the features used by our detector to be computed very quickly. In order to achieve true scale invariance, almost all face detection systems must operate on multiple image scales. The integral image, by eliminating the need to compute a multi-scale image pyramid, reduces significantly. Using the integral image, face detection is completed in almost the same time as it takes for an image pyramid to be computed. The initial image processing required for face detection. The second contribution of this paper is a simple and efficient classifier built from computationally efficient features using AdaBoost for feature selection. This classifier is clearly an effective one for face detection and the authors are confident that it will also be effective in other domains such as automobile or pedestrian detection. Furthermore, the idea of an aggressive and effective technique for feature selection should have impact on a wide variety of learning tasks. The third contribution is a method for combining classifiers in a “cascade” which allows background regions of the image to be quickly discarded while spending more computation on promising face-like regions.

**Advantage:**

* + Minimizes computation time while achieving high detection accuracy and effectiveness.
  + Helps in very fast feature evaluation.
  + Is a simple and efficient classifier that is built by selecting a small number of important features from a huge library of potential features using AdaBoost.
  + Reduces computation time while improving detection accuracy.

**Disadvantage:**

* + The proposed system uses large and complex dataset which are difficult and time consuming.

##### Title: Vision-Based Sign Language Translation Device

***Author:*** *Yellapu Madhuri, Anitha.G, Anburajan.M*

This report presents a mobile VISION-BASEDSIGN LANGUAGE TRANSLATION DEVICE for automatic translation of Indian sign language into speech in English to assist the hearing and/or speech impaired people to communicate with hearing people. The authors proposed a real-time vision-based system for recognizing finger spelling continuous Sign Language (ASL) using a single camera to track the user's unadorned hands. This system is broken down into three main parts starting with the image acquisition followed by image processing to extract features for recognition and last comes the recognition stage where signs are identified and audio output is given. The program starts with image acquisition,

i.e. sign images capturing by the camera. The acquired images are pre-processed to differentiate static and dynamic signs, and also the start and end of a sign. The images are processed to identify the region of interest. The unique features of each sign in the region of interest are extracted to be used in the recognition stage. In the recognition stage, the features extracted are compared with the available database of pattern matching templates. A threshold value is set for the maximum difference between the input sign and the database, if the difference is below the maximum limit, a match is found and the sign is recognized. Corresponding audio file is played on audio device. The program can be implemented in a laptop, desktop or an IOS mobile phone to operate with its inbuilt camera, processor and audio device.

**Advantage:**

* + It can be used as a translator between deaf and people that do not understand sign language, avoiding by this way the intervention of an intermediate person.
  + The proposed system is highly consistent, reproducible, with fairly high precision and accuracy.

**Disadvantage**:

* + This project did not focus on facial expressions although it is well known that facial expressions convey important part of sign-languages.

1. **Title: Survey on Various Gesture Recognition Techniques For Interfacing Machines Based On Ambient Intelligence**

***Author: Harshith.C, Karthik.R.Shastry, Manoj Ravindran, M.V.V.N.S Srikanth, Naveen Lakshmikhanth***

Gesture recognition is mainly apprehensive on analyzing the functionality of human wits. The main goal of gesture recognition is to create a system which can recognize specific human gestures and use them to convey information or for device control. Hand gestures provide a separate complementary modality to speech for expressing ones ideas. Information associated with hand gestures in a conversation is degree, discourse structure, spatial and temporal structure. The approaches present can be mainly divided into Data- Glove Based and Vision Based approaches. An important face feature point is the nose tip. Since nose is the highest protruding point from the face. Besides that, it is not affected by facial expressions. Another important function of the nose is that it is able to indicate the head pose. Knowledge of the nose ocation will enable us to align an unknown 3D face with those in a face database. Eye detection is divided into eye position detection and eye contour detection. The purpose of this paper is to compare various human Gesture recognition systems for interfacing machines directly to human wits without any corporeal media in an ambient environment.

**Advantage:**

* + By incorporating the proposed method, the efficiency could be enhanced to greater extent.

**Disadvantage:**

* + The devices are quite expensive and bring much cumbersome experience to the users.

##### Title: A New 2D Static Hand Gesture Color Image Dataset for ASL Gestures

***Author:*** A.L.C. Barczak, N.H. Reyes, M. Abastillas, A. Piccio and T. Susnjak

It usually takes a fusion of image processing and machine learning algorithms in order to build a fully-functioning computer vision system for hand gesture recognition.

Fortunately, the complexity of developing such a system could be alleviated by treating the system as a collection of multiple sub-systems working together, in such a way that they can be dealt with in isolation. Machine learning need to feed on thousands of exemplars (e.g. images, features) to automatically establish some recognizable patterns for all possible classes (e.g. and gestures) that applies to the problem domain. A good number of exemplars helps, but it is also important to note that the efficacy of these exemplars depends on the variability of illumination conditions, hand postures, angles of rotation, scaling and on the number of volunteers from whom the hand gesture images were taken. These exemplars are usually subjected to image processing first, to reduce the presence of noise and extract the important features from the images. These features serve as inputs to the machine learning system. Different sub-systems are integrated together to form a complete computer vision system for gesture recognition. The main contribution of this work is on the production of the exemplars. A minor contribution is given in the form of a specific feature extraction method called moment invariants, for which the computation method and the values are furnished with the dataset.

**Advantage:**

* + For gestures, no need to use special gloves, or any other apparatus.

**Disadvantage:**

* + Images were taken at a certain angle of rotation (perpendicular to the subject), which limits the number of samples.

#### Chapter 3

**3.1 SYSTEM STUDY**

**3.1.1 FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

##### ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

##### TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

##### SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

##### Functional Requirement

* Laptop can capture live video using the web camera
* The captured video is processed and features are extracted
* Can detect the signs
* Application can send the instruction to the machine learning model
* can process the instruction model
* The laptop’s current screen image is captured
* The sign status is displayed in the Indian sign Language

##### Non-Functional Requirement

* **Usability**

The client acknowledges be typical nearly the buyer interfaces and committed to ask for ambush pressure in relocating to a unique framework with another condition.

##### Reliability

The progressions made by the Programmer ought to be obvious both to the Project pioneer and in addition the Test design.

##### Security

Counting bug following the framework must give important security and must secure the entire procedure from slamming.

##### Performance

The framework will be facilitated on a solitary web server with a solitary database server out of sight, consequently execution turns into a noteworthy concern.

##### Portability

This is required when the web server, which is facilitating the framework stalls out because of a few issues, which requires their framework to be taken to another framework.

##### Reusability

The framework ought to be separated into such modules that it could be utilized as a piece of another framework without requiring a lot of work.3.6 Technologies used.

#### Chapter 4

##### SYSTEM REQUIREMENTS SPECIFICATION

* + 1. **Requirement specification**

Framework Requirement Specification (SRS) is a focal report, which outlines the foundation of the item headway handle. It records the necessities of a structure and in addition has a delineation of its noteworthy highlight. An SRS is basically an affiliation's seeing (in making) of a customer or potential client's edge work necessities and conditions at a particular point in time (for the most part) before any veritable design or change work. It's a two-way insurance approach that ensures that both the client and the affiliation understand exchange's necessities from that perspective at a given point in time.

The synthesis of programming need detail reduces headway effort, as careful review of the report can reveal oversights, mixed up presumptions, and inconsistencies in front of plan for the change cycle when these issues are less requesting to right. The SRS discusses the thing however not the wander that made it, thusly the SRS fills in as a start for later change of the finished thing.

The SRS may should be changed, be that as it may it gives a foundation to continue with creation appraisal. In direct words, programming need assurance is the starting phase of the item change activity. The SRS implies unraveling the musings in the brains of the clients – the data, into a formal chronicle – the yield of the essential stage. In this manner the yield of the stage is an arranged of formally decided necessities, which in a perfect world are done and relentless, while the information has none of these properties.

##### Hardware Requirements:

Processor : Any Processor above 500 MHz

RAM : 2 GB

Hard Disk : 80 GB

##### Software Requirements:

1. Operating system : Windows 7/8/10
2. IDE : Python, Anaconda Navigator
3. Programming Language: Python

##### SOFTWARE DESCRIPTION PYTHON

Python is an object-oriented programming language created by Guido

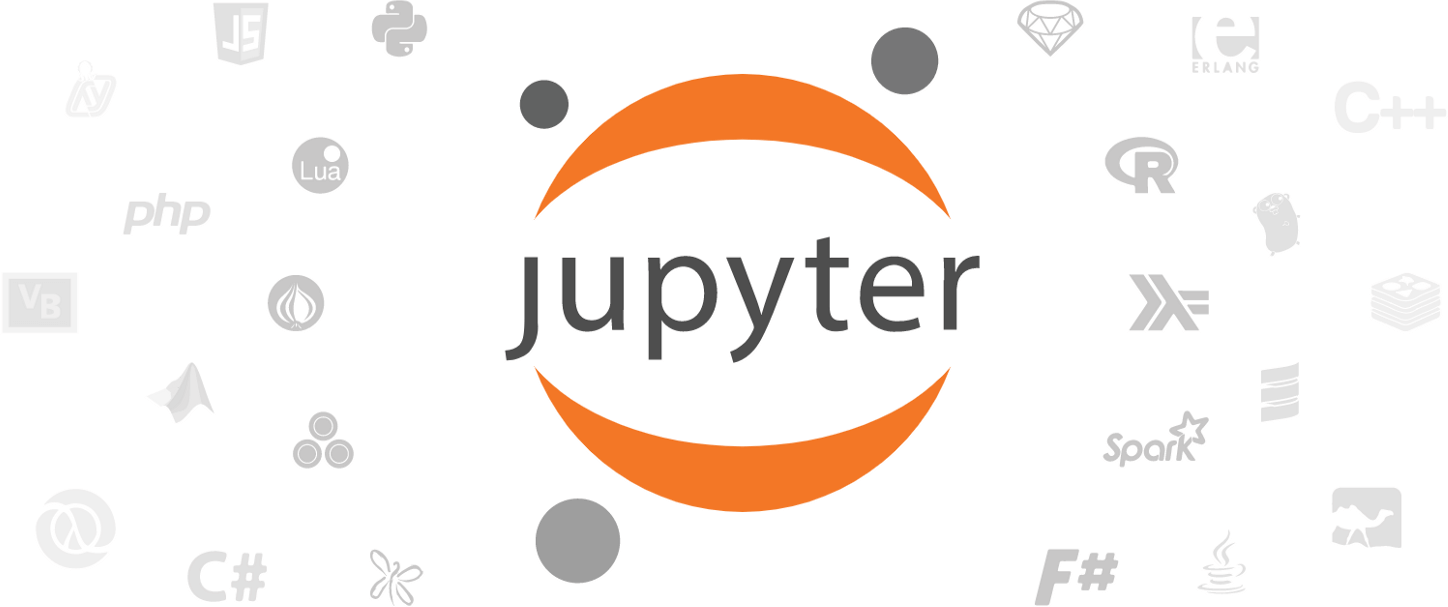
Rossum in 1989. It is ideally designed for rapid prototyping of complex applications. It has interfaces to many OS system calls and libraries and is extensible to C or C++. Many large companies use the Python programming language include NASA, Google, YouTube, Bit Torrent, etc.

Python is widely used in Artificial Intelligence, Natural Language Generation, Neural Networks and other advanced fields of Computer Science. Python had deep focus on code readability & this class will teach you python from basics.

##### Characteristics of Python

* It provides rich data types and easier to read syntax than any other programming languages
* It is a platform independent scripted language with full access to operating system API's
* Compared to other programming languages, it allows more run-time flexibility
* It includes the basic text manipulation facilities of Perl and Awk
* A module in Python may have one or more classes and free functions
* Libraries in Pythons are cross-platform compatible with Linux, MacIntosh, and Windows

##### Jupyter Notebook:



The Jupyter Notebook is an incredibly powerful tool for interactively developing and presenting data science projects. This article will walk you through how to set up Jupyter Notebooks on your local machine and how to start using it to do data science projects.

First, though: what is a “notebook”? A notebook integrates code and it’s output into a single document that combines visualizations, narrative text, mathematical equations, and other rich media. This intuitive workflow promotes iterative and rapid development, making notebooks an increasingly popular choice at the heart of contemporary data science, analysis, and increasingly science at large.

Best of all, as part of the open source [Project Jupyter](https://jupyter.org/), they are completely free.

The Jupyter project is the successor to the earlier I Python Notebook, which was first published as a prototype in 2010. Although it is possible to use many different programming languages within Jupyter Notebooks, this article will focus on Python as it is the most common use case. (Among R users, [R](https://www.rstudio.com/) [Studio](https://www.rstudio.com/) tends to be a more popular choice).

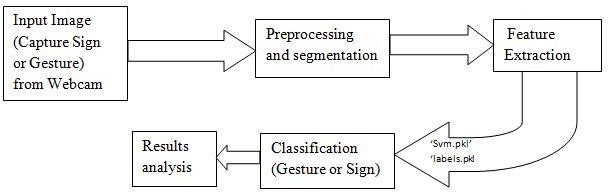
To get the most out of this tutorial you should be familiar with programming, specifically Python and [pandas](https://pandas.pydata.org/) specifically. That said, if you have experience with another language, the Python in this article shouldn’t be too cryptic, and will still help you get Jupyter Notebooks set up locally. Jupyter Notebooks can also act as a flexible platform for getting to grips with pandas and even Python, as will become apparent in this article.

We will:

* Cover the basics of installing Jupyter and creating your first notebook
* Delve deeper and learn all the important terminology
* Explore how easily notebooks can be shared and published online. Indeed, this article *is* a Jupyter Notebook! Everything here was written in the Jupyter Notebook environment, though you are viewing it in a read-only form.

##### Modules

The proposed system consists are five modules of following steps to interpret the gesture from the input image such as:



##### Fig 1: Overview of the proposed system in five modules

* **Input Image (Capture sign or Gesture) from Webcam:**

The image (gesture or sign) is captured using the laptop camera or the external device webcam to get better image clarity.

##### Preprocessing and segmentation:

Image processing is necessary for image enhancement. During Preprocessing RGB image to convert into HSV color space. This step was taken because HSV color space was less sensitive to illumination changes compared to RGB. Then it was filtered, smoothened and finally the biggest binary linked object was being considered so as to avoid consideration of skin-colored objects other than hand. To obtains the good result smoothing and filtering is done. Image segmentation is basically performed to locate the hand object in image.

##### Feature Extraction:

Feature Extraction stage is necessary because certain features have to be extracted so that they are unique for each gesture or sign. After the decision is made that a sign is present, then the last frame is taken into consideration and features. The Feature Extraction is extracting the features in all Images (gesture or sign) dataset are store in ‘svm.pkl' and finally extract the labels store in ‘labels.pkl' best on train data and test data.

##### Classification:

Classification of hand is done with the help of various features calculated previously. The five-bit binary sequence is thus generated to uniquely recognize and utilize these recognized the recognized hand gesture for supporting human computer interaction. By the feature extraction significant peak is encoded as 1 while insignificant peak is encoded as 0 based on intersection to the threshold line.

##### Results analysis:

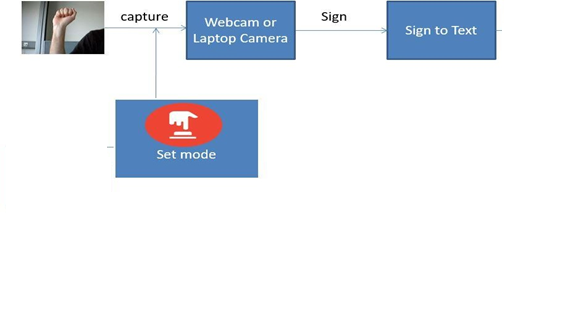
Different images were tested and found that the new technique of classification was found to show 97% accuracy. Some images tested with other database images are given in the results analysis. In **Results analysis** are real time detect the sign language and sign recognize when live camera is start then capture the test images (gesture or sign) that time compare the features ‘svm.pkl' and ‘labels.pkl' if it is matching the dataset after the process in display the result.

##### SYSTEM DESIGN

* + 1. **SYSTEM ARCHITECTURE**

The architectural configuration procedure is concerned with building up a fundamental basic system for a framework. It includes recognizing the real parts of the framework and interchanges between these segments. The beginning configuration procedure of recognizing these subsystems and building up a structure for subsystem control and correspondence is called

construction modeling outline and the yield of this outline procedure is a portrayal of the product structural planning. The proposed architecture for this system is given below. It shows the way this system is designed and brief working of the system.



##### Fig 2: Overview of the High-Level Block Diagram.

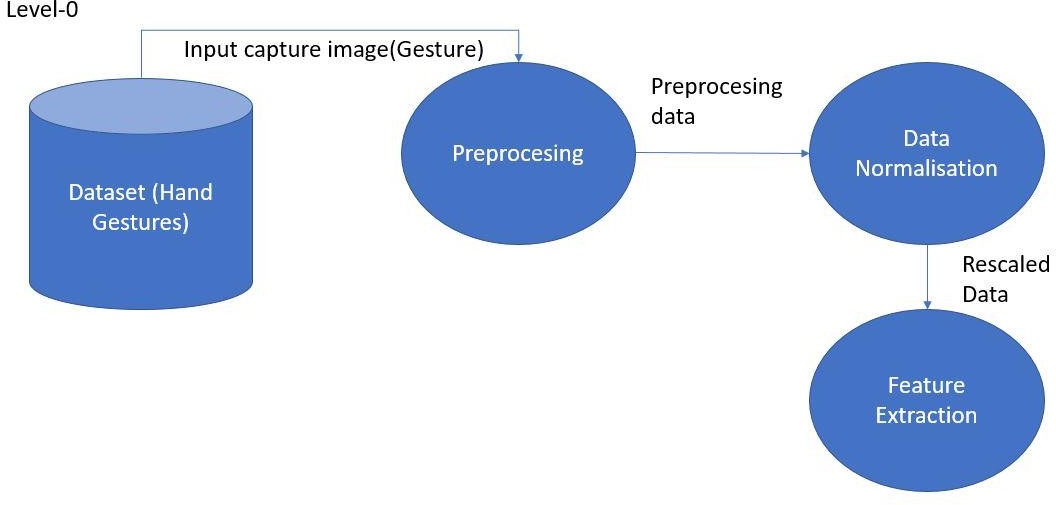
* + 1. **Data Flow Diagram**

The DFD is straightforward graphical formalism that can be utilized to speak to a framework as far as the info information to the framework, different preparing did on this information and the yield information created by the framework. A DFD model uses an exceptionally predetermined number of primitive images to speak to the capacities performed by a framework and the information stream among the capacities.

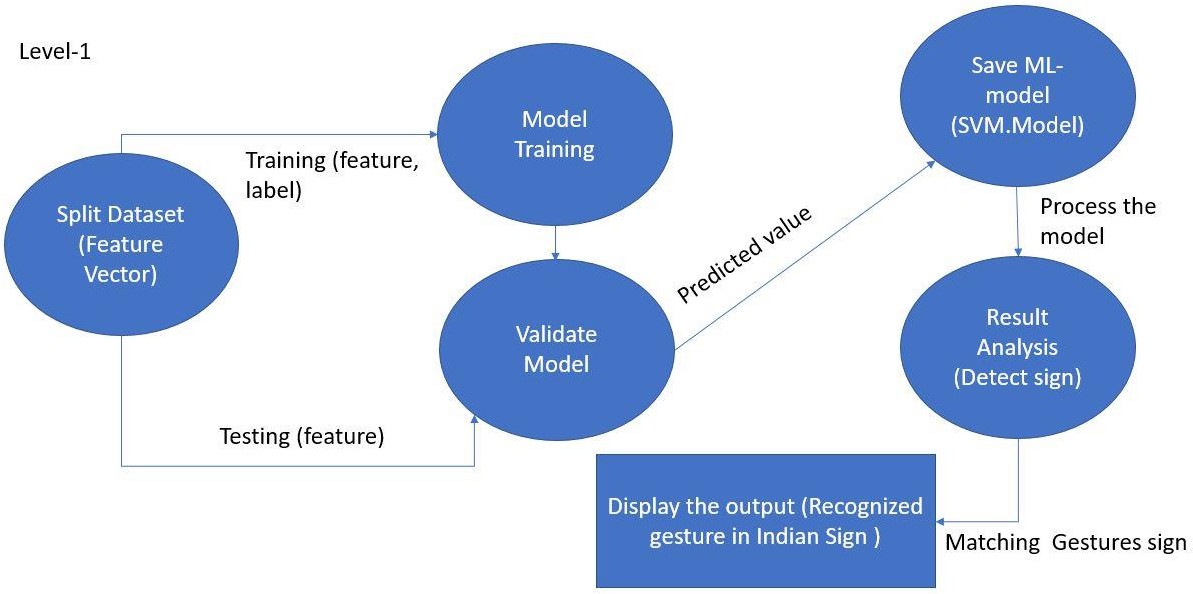
The principal motivation behind why the DFD method is so famous is most likely in light of the way that DFD is an exceptionally basic formalism- It is easy to comprehend and utilization. Beginning with the arrangement of abnormal state works that a framework performs, a DFD display progressively speaks to different sub capacities. Actually, any various leveled model is easy to get it.

The human personality is such that it can without much of a stretch see any progressive model of a framework in light of the fact that in a various leveled model, beginning with an extremely straightforward and unique model of framework, distinctive points of interest of a framework are gradually presented through the diverse orders. A data-flow diagram (DFD) is a graphical representation of the "stream" of information through a data framework. DFDs can likewise be utilized for the perception of information handling.

Level-0



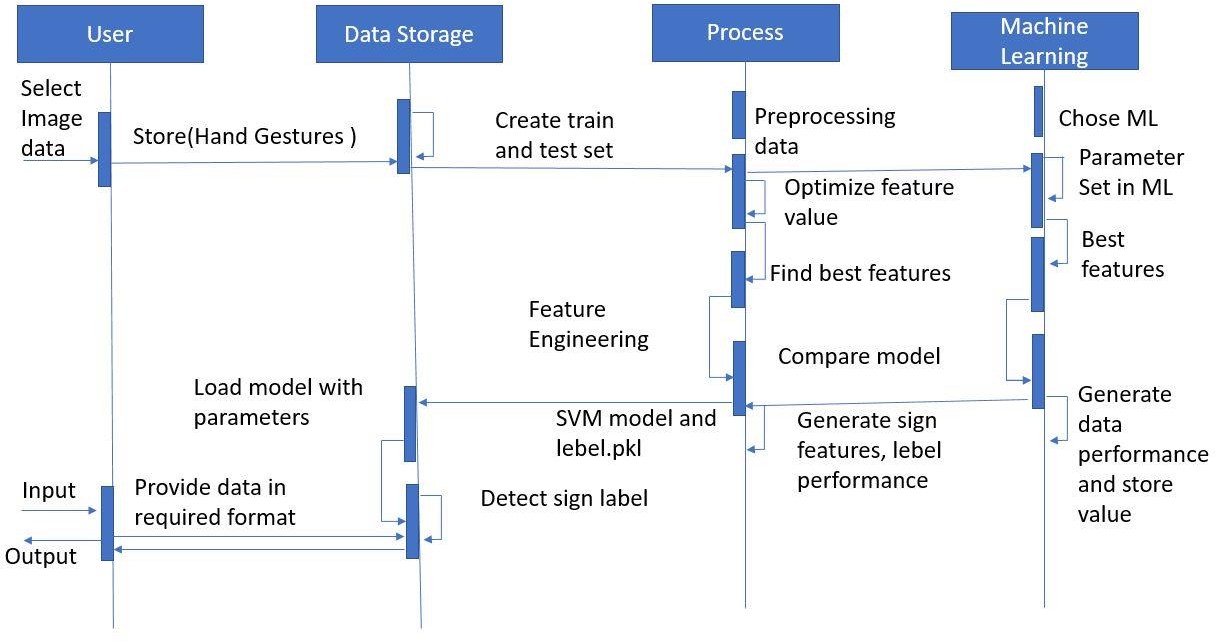
Level-1



**Fig 3: Image Data Flow Diagram.**

* + 1. **Sequence Diagram**

A sequence diagram is an integrated Modelling Language is a sort of communication diagram that shows procedures work with each other and in what request. Sequence diagrams are some of the times called occasion follow diagrams, occasion situations, and timing diagram. Sequence diagrams are utilized to formalize the conduct of the framework and to picture the correspondence among articles. They are valuable for recognizing extra questions that takes part in the utilization cases. A sequence diagram speaks to the associations that happen among these articles.

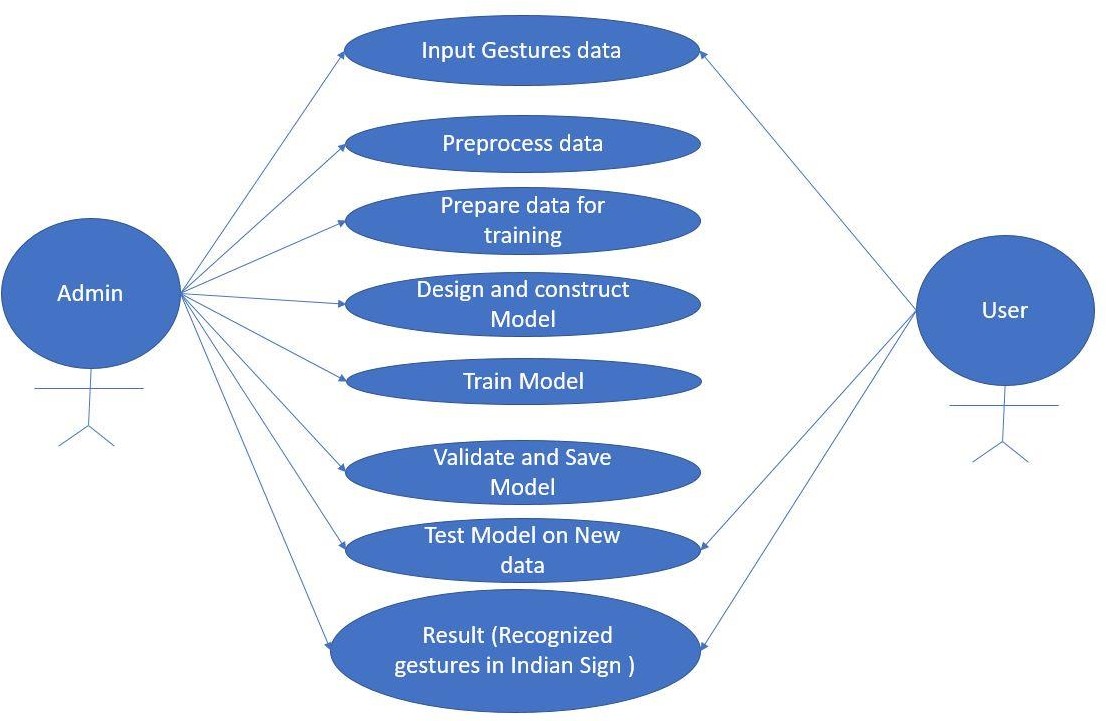


##### Fig 4: Overview of Sequence Diagram.

* + 1. **Use case Diagram**

A use case chart is a kind of behavioral graph made from a Use-case examination. Its object is to present a graphical diagram of the usefulness gave by a framework regarding performers, their objectives (spoke to as utilization cases), and any conditions between those utilization cases. Use case chart gives us the data about how that clients and utilization cases are connected with the framework. Use cases are used amid prerequisites elicitation and examination to speak to the usefulness of the framework. Use cases concentrate on the conduct of the framework from an outside perspective.

A use case depicts a capacity gave by framework that yields an obvious result for a performer. A performing artist portrays any element that collaborates with the system. The performers are outside the limit of the framework, while the use cases are inside the limit of the framework. On-screen characters are spoken to with stick figures, use cases with ovals, and the limit of the framework with a container encasing the use cases.

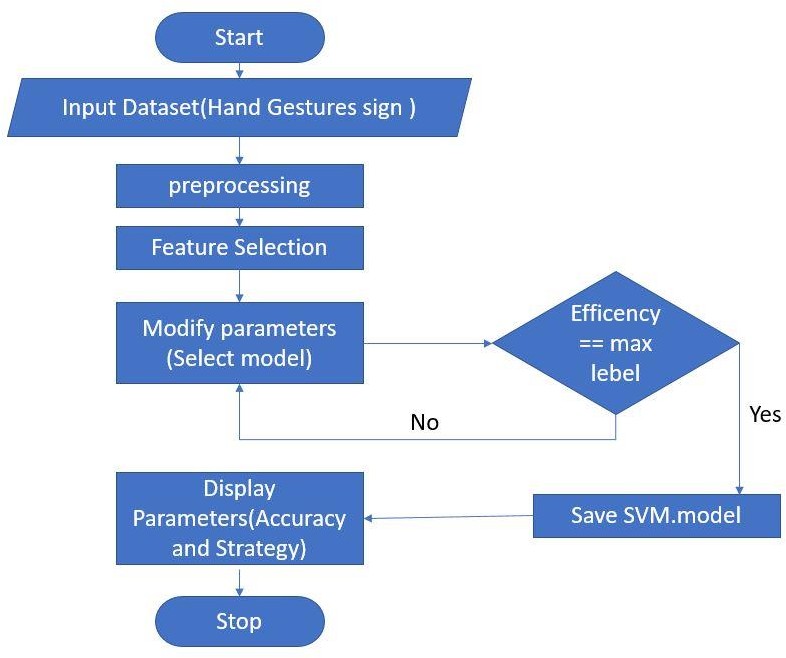


**Fig 5: Overview of Use Case Diagram.**

* + 1. **Flow chart**

A flowchart is a type of [diagram](https://en.wikipedia.org/wiki/Diagram) that represents a [workflow](https://en.wikipedia.org/wiki/Workflow) or [process](https://en.wikipedia.org/wiki/Process). A flowchart can also be defined as a diagrammatic representation of an [algorithm](https://en.wikipedia.org/wiki/Algorithm), a step-by-step approach to solving a task.

The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given [problem](https://en.wikipedia.org/wiki/Problem_solving). Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.



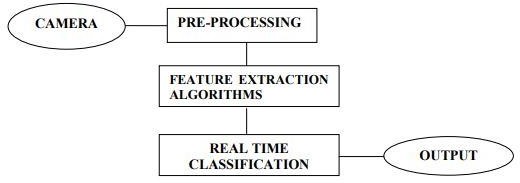
**Fig 6: Flow Chart.**

#### Chapter 5

##### Implementation

**Hand gesture recognition can be divided into three phases:**

* + - * Preprocessing.
      * Feature extraction of processed image.
      * Real time classification.



##### Fig 7: System Implementation

* + 1. **PRE-PROCESSING**

Like many other pattern recognition tasks, pre-processing is necessary for enhancing robustness and recognition accuracy.

The preprocessing prepares the image sequence for the recognition, so before calculating the diagonal Sum and other algorithms, a pre-processing step is performed to get the appropriate image, which is required for real time classification. So, it consists of some steps. The net effect of this processing is to extract the hand only from the given input because once the

hand is detected from the given input it can be recognized easily. So pre- processing step mainly consists of following tasks:

* + - * Skin Modeling
      * Removal of Background
      * Conversion from RGB to binary
      * Hand Detection

##### Skin Modelling

There are numerous methods used for skin detection such as RGB (Red, Green, Blue), YCbCr (Luminance Chrominance) and HSV (Hue, Saturation, Value).

##### RGB:

RGB is a 3D color space pixel where each pixel has combination of three colors Red, Green and Blue at specific location. This technique widely used in image processing for identifying skin region.

##### HSV (Hue, Saturation and Value):

In HSV, Hue detect dominant color and Saturation define colorfulness whilst Value measure intensity or brightness. This is well enough to choose single color but it ignores complexity of color appearance. It trades off computation speed mean computationally expensive and perceptual relevance.

##### Skin Detection

The skin color detection is one of important goal in hand gesture recognition. Skin color detection decision rules which we have to build that will discriminate between skin portion and non-skin portion pixels. This is accomplished usually by metric introduction, which measure distance of the pixel color. This metric type is knows as skin modelling.

##### Removal of Background

I have found that background greatly affects the results of hand detection that’s why I have decided to remove it. For this I have written our own code in spite of using any built-in ones.

##### Conversion from RGB to Binary

All algorithms accept an input in RGB form and then convert it into binary format in order to provide ease in recognizing any gesture and also retaining the luminance factor in an image.

##### Hand detection

Image could have more than one skin area but we required only hand for further process. For this I choose criteria image labeling which is following:

#### FEATURE EXTRACTION OF PROCESSED IMAGE.

##### Noise reduction

Once the hand gesture image has been segmented, a special filter is applied to remove noise by eliminating all the single white pixels on a black background and all the single black pixels on a white foreground. To accomplish this goal, a median filter is applied to the segmented image.

##### Skin Mask

This algorithm makes conversion from colorful image to binary skin mask. We use color components to make proper decision is the current pixel fall down to skin color space or not. This results in a binary image called skin mask.

#### CLASSIFICATION

In a typical classification system image is captured by a camera and consequently processed. In Supervised classification, first of all training took place through known group of pixels.

The classifiers are SVM, KNeighbors Classifier, Logistic Regression, MLP Classifier, Naïve Bayes and Random Forest Classifier.

##### Coding

**Code for Capture**

import cv2

import time

import numpy as np

import os

def nothing(x):

pass

image\_x, image\_y = 64, 64

def create\_folder(folder\_name):

if not os.path.exists('./mydata/training\_set/' + folder\_name):

os.mkdir('./mydata/training\_set/' + folder\_name)

if not os.path.exists('./mydata/test\_set/' + folder\_name):

os.mkdir('./mydata/test\_set/' + folder\_name)

def capture\_images(ges\_name):

create\_folder(str(ges\_name))

cam = cv2.VideoCapture(0)

cv2.namedWindow("test")

img\_counter = 0

t\_counter = 1

training\_set\_image\_name = 1

test\_set\_image\_name = 1

listImage = [1,2,3,4,5]

cv2.namedWindow("Trackbars")

cv2.createTrackbar("L - H", "Trackbars", 0, 179, nothing)

cv2.createTrackbar("L - S", "Trackbars", 0, 255, nothing)

cv2.createTrackbar("L - V", "Trackbars", 0, 255, nothing)

cv2.createTrackbar("U - H", "Trackbars", 179, 179, nothing)

cv2.createTrackbar("U - S", "Trackbars", 255, 255, nothing)

cv2.createTrackbar("U - V", "Trackbars", 255, 255, nothing)

for loop in listImage:

while True:

ret, frame = cam.read()

frame = cv2.flip(frame, 1)

l\_h = cv2.getTrackbarPos("L - H", "Trackbars")

l\_s = cv2.getTrackbarPos("L - S", "Trackbars")

l\_v = cv2.getTrackbarPos("L - V", "Trackbars")

u\_h = cv2.getTrackbarPos("U - H", "Trackbars")

u\_s = cv2.getTrackbarPos("U - S", "Trackbars")

u\_v = cv2.getTrackbarPos("U - V", "Trackbars")

img = cv2.rectangle(frame, (425, 100), (625, 300), (0, 255, 0), thickness=2, lineType=8, shift=0)

lower\_blue = np.array([l\_h, l\_s, l\_v])

upper\_blue = np.array([u\_h, u\_s, u\_v])

imcrop = img[102:298, 427:623]

hsv = cv2.cvtColor(imcrop, cv2.COLOR\_BGR2HSV)

mask = cv2.inRange(hsv, lower\_blue, upper\_blue)

result = cv2.bitwise\_and(imcrop, imcrop, mask=mask)

cv2.putText(frame, str(img\_counter), (30, 400), cv2.FONT\_HERSHEY\_TRIPLEX, 1.5, (127, 127, 255))

cv2.imshow("test", frame)

cv2.imshow("mask", mask)

cv2.imshow("result", result)

if cv2.waitKey(1) == ord('c'):

if t\_counter <= 350:

img\_name = "./mydata/training\_set/" + str(ges\_name) + "/{}.png".format(training\_set\_image\_name)

save\_img = cv2.resize(mask, (image\_x, image\_y))

cv2.imwrite(img\_name, save\_img)

print("{} written!".format(img\_name))

training\_set\_image\_name += 1

if t\_counter > 350 and t\_counter <= 400:

img\_name = "./mydata/test\_set/" + str(ges\_name) + "/{}.png".format(test\_set\_image\_name)

save\_img = cv2.resize(mask, (image\_x, image\_y))

cv2.imwrite(img\_name, save\_img)

print("{} written!".format(img\_name))

test\_set\_image\_name += 1

if test\_set\_image\_name > 250:

break

t\_counter += 1

if t\_counter == 401:

t\_counter = 1

img\_counter += 1

elif cv2.waitKey(1) == 27:

break

if test\_set\_image\_name > 250:

break

cam.release()

cv2.destroyAllWindows()

ges\_name = input("Enter gesture name: ")

capture\_images(ges\_name)

**Code for cnn\_model**

# Part 1 - Building the CNN

#importing the Keras libraries and packages

from keras.models import Sequential

from keras.layers import Convolution2D

from keras.layers import MaxPooling2D

from keras.layers import Flatten

from keras.layers import Dense, Dropout

from keras import optimizers

# Initialing the CNN

classifier = Sequential()

# Step 1 - Convolutio Layer

classifier.add(Convolution2D(32, 3, 3, input\_shape = (64, 64, 3), activation = 'relu'))

#step 2 - Pooling

classifier.add(MaxPooling2D(pool\_size =(2,2)))

# Adding second convolution layer

classifier.add(Convolution2D(32, 3, 3, activation = 'relu'))

classifier.add(MaxPooling2D(pool\_size =(2,2)))

#Adding 3rd Concolution Layer

classifier.add(Convolution2D(64, 3, 3, activation = 'relu'))

classifier.add(MaxPooling2D(pool\_size =(2,2)))

#Step 3 - Flattening

classifier.add(Flatten())

#Step 4 - Full Connection

classifier.add(Dense(256, activation = 'relu'))

classifier.add(Dropout(0.5))

classifier.add(Dense(26, activation = 'softmax'))

#Compiling The CNN

classifier.compile(

optimizer = optimizers.SGD(lr = 0.01),

loss = 'categorical\_crossentropy',

metrics = ['accuracy'])

#Part 2 Fittting the CNN to the image

from keras.preprocessing.image import ImageDataGenerator

train\_datagen = ImageDataGenerator(

rescale=1./255,

shear\_range=0.2,

zoom\_range=0.2,

horizontal\_flip=True)

test\_datagen = ImageDataGenerator(rescale=1./255)

training\_set = train\_datagen.flow\_from\_directory(

'mydata/training\_set',

target\_size=(64, 64),

batch\_size=32,

class\_mode='categorical')

test\_set = test\_datagen.flow\_from\_directory(

'mydata/test\_set',

target\_size=(64, 64),

batch\_size=32,

class\_mode='categorical')

model = classifier.fit\_generator(

training\_set,

steps\_per\_epoch=800,

epochs=25,

validation\_data = test\_set,

validation\_steps = 6500

)

'''#Saving the model

import h5py

classifier.save('Trained\_model.h5')'''

print(model.history.keys())

import matplotlib.pyplot as plt

# summarize history for accuracy

plt.plot(model.history['acc'])

plt.plot(model.history['val\_acc'])

plt.title('model accuracy')

plt.ylabel('accuracy')

plt.xlabel('epoch')

plt.legend(['train', 'test'], loc='upper left')

plt.show()

# summarize history for loss

plt.plot(model.history['loss'])

plt.plot(model.history['val\_loss'])

plt.title('model loss')

plt.ylabel('loss')

plt.xlabel('epoch')

plt.legend(['train', 'test'], loc='upper left')

plt.show()

**Code to Recognise**

import cv2

import numpy as np

def nothing(x):

pass

image\_x, image\_y = 64,64

from keras.models import load\_model

classifier = load\_model('Trained\_model.h5')

def predictor():

import numpy as np

from keras.preprocessing import image

test\_image = image.load\_img('1.png', target\_size=(64, 64))

test\_image = image.img\_to\_array(test\_image)

test\_image = np.expand\_dims(test\_image, axis = 0)

result = classifier.predict(test\_image)

if result[0][0] == 1:

return 'A'

elif result[0][1] == 1:

return 'B'

elif result[0][2] == 1:

return 'C'

elif result[0][3] == 1:

return 'D'

elif result[0][4] == 1:

return 'E'

elif result[0][5] == 1:

return 'F'

elif result[0][6] == 1:

return 'G'

elif result[0][7] == 1:

return 'H'

elif result[0][8] == 1:

return 'I'

elif result[0][9] == 1:

return 'J'

elif result[0][10] == 1:

return 'K'

elif result[0][11] == 1:

return 'L'

elif result[0][12] == 1:

return 'M'

elif result[0][13] == 1:

return 'N'

elif result[0][14] == 1:

return 'O'

elif result[0][15] == 1:

return 'P'

elif result[0][16] == 1:

return 'Q'

elif result[0][17] == 1:

return 'R'

elif result[0][18] == 1:

return 'S'

elif result[0][19] == 1:

return 'T'

elif result[0][20] == 1:

return 'U'

elif result[0][21] == 1:

return 'V'

elif result[0][22] == 1:

return 'W'

elif result[0][23] == 1:

return 'X'

elif result[0][24] == 1:

return 'Y'

elif result[0][25] == 1:

return 'Z'

cam = cv2.VideoCapture(0)

cv2.namedWindow("Trackbars")

cv2.createTrackbar("L - H", "Trackbars", 0, 179, nothing)

cv2.createTrackbar("L - S", "Trackbars", 0, 255, nothing)

cv2.createTrackbar("L - V", "Trackbars", 0, 255, nothing)

cv2.createTrackbar("U - H", "Trackbars", 179, 179, nothing)

cv2.createTrackbar("U - S", "Trackbars", 255, 255, nothing)

cv2.createTrackbar("U - V", "Trackbars", 255, 255, nothing)

cv2.namedWindow("test")

img\_counter = 0

img\_text = ''

while True:

ret, frame = cam.read()

frame = cv2.flip(frame,1)

l\_h = cv2.getTrackbarPos("L - H", "Trackbars")

l\_s = cv2.getTrackbarPos("L - S", "Trackbars")

l\_v = cv2.getTrackbarPos("L - V", "Trackbars")

u\_h = cv2.getTrackbarPos("U - H", "Trackbars")

u\_s = cv2.getTrackbarPos("U - S", "Trackbars")

u\_v = cv2.getTrackbarPos("U - V", "Trackbars")

img = cv2.rectangle(frame, (425,100),(625,300), (0,255,0), thickness=2, lineType=8, shift=0)

lower\_blue = np.array([l\_h, l\_s, l\_v])

upper\_blue = np.array([u\_h, u\_s, u\_v])

imcrop = img[102:298, 427:623]

hsv = cv2.cvtColor(imcrop, cv2.COLOR\_BGR2HSV)

mask = cv2.inRange(hsv, lower\_blue, upper\_blue)

cv2.putText(frame, img\_text, (30, 400), cv2.FONT\_HERSHEY\_TRIPLEX, 1.5, (0, 255, 0))

cv2.imshow("test", frame)

cv2.imshow("mask", mask)

#if cv2.waitKey(1) == ord('c'):

img\_name = "1.png"

save\_img = cv2.resize(mask, (image\_x, image\_y))

cv2.imwrite(img\_name, save\_img)

print("{} written!".format(img\_name))

img\_text = predictor()

if cv2.waitKey(1) == 27:

break

cam.release()

cv2.destroyAllWindows()

**Code to Test Module**

from keras.models import load\_model

classifier = load\_model('Trained\_model.h5')

classifier.evaluate()

#Prediction of single image

import numpy as np

from keras.preprocessing import image

img\_name = input('Enter Image Name: ')

image\_path = './predicting\_data/{}'.format(img\_name)

print('')

test\_image = image.load\_img(image\_path, target\_size=(200, 200))

test\_image = image.img\_to\_array(test\_image)

test\_image = np.expand\_dims(test\_image, axis = 0)

result = classifier.predict(test\_image)

#training\_set.class\_indices

print('Predicted Sign is:')

print('')

if result[0][0] == 1:

print('A')

elif result[0][1] == 1:

print('B')

elif result[0][2] == 1:

print('C')

elif result[0][3] == 1:

print('D')

elif result[0][4] == 1:

print('E')

elif result[0][5] == 1:

print('F')

elif result[0][6] == 1:

print('G')

elif result[0][7] == 1:

print('H')

elif result[0][8] == 1:

print('I')

elif result[0][9] == 1:

print('J')

elif result[0][10] == 1:

print('K')

elif result[0][11] == 1:

print('L')

elif result[0][12] == 1:

print('M')

elif result[0][13] == 1:

print('N')

elif result[0][14] == 1:

print('O')

elif result[0][15] == 1:

print('P')

elif result[0][16] == 1:

print('Q')

elif result[0][17] == 1:

print('R')

elif result[0][18] == 1:

print('S')

elif result[0][19] == 1:

print('T')

elif result[0][20] == 1:

print('U')

elif result[0][21] == 1:

print('V')

elif result[0][22] == 1:

print('W')

elif result[0][23] == 1:

print('X')

elif result[0][24] == 1:

print('Y')

elif result[0][25] == 1:

print('Z')

#### Chapter 6

##### Testing and Types

**6.1.1 Testing:**

Testing is an important phase in the development life cycle of the product. This is the phase, where the remaining errors, if any, from all the phases are detected. Hence testing performs a very critical role for quality assurance and ensuring the reliability of the software.

During the testing, the program to be tested was executed with a set of test cases and the output of the program for the test cases was evaluated to determine whether the program was performing as expected. Errors were found and corrected by using the below stated testing steps and correction was recorded for future references. Thus, a series of testing was performed on the system, before it was ready for implementation.

It is the process used to help identify the correctness, completeness, security, and quality of developed computer software. Testing is a process of technical investigation, performed on behalf of stake holders, i.e. intended to reveal the quality-related information about the product with respect to context in which it is intended to operate. This includes, but is not limited to, the process of executing a program or application with the intent of finding errors.

The quality is not an absolute; it is value to some person. With that in mind, testing can never completely establish the correctness of arbitrary computer software; Testing furnishes a ‘criticism’ or comparison that compares the state and behavior of the product against specification. An important point

is that software testing should be distinguished from the separate discipline of Software Quality Assurance (SQA), which encompasses all business process areas, not just testing.

There are many approaches to software testing, but effective testing of complex products is essentially a process of investigation not merely a matter of creating and following routine procedure.

Although most of the intellectual processes of testing are nearly identical to that of review or inspection, the word testing is connoted to mean the dynamic analysis of the product-putting the product through its paces. Some of the common quality attributes include capability, reliability, efficiency, portability, maintainability, compatibility and usability.

A good test is sometimes described as one, which reveals an error; however, more recent thinking suggest that a good test is one which reveals information of interest to someone who matters within the project community.

##### Types of Testing

* + 1. **Unit Testing**

Individual component are tested to ensure that they operate correctly. Each component is tested independently, without other system component. This system was tested with the set of proper test data for each module and the results were checked with the expected output. Unit testing focuses on verification effort on the smallest unit of the software design module. This is also known as MODULE TESTING. This testing is carried out during phases,

each module is found to be working satisfactory as regards to the expected

output from the module.

##### Integration Testing

Integration testing is another aspect of testing that is generally done in order to uncover errors associated with flow of data across interfaces. The unit- tested modules are grouped together and tested in small segment, which make it easier to isolate and correct errors. This approach is continued unit I have integrated all modules to form the system as a whole.

##### System Testing

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration testing. System testing is based on process description and flows, emphasizing pre-driver process and integration points.

##### Performance Testing

The performance testing ensure that the output being produced within the time limits and time taken for the system compiling, giving response to the users and request being send to the system in order to retrieve the results.

##### Validation Testing

The validation testing can be defined in many ways, but a simple definition is that. Validation succeeds when the software functions in a manner that can be reasonably expected by the end user.

##### Black Box testing

Black box testing is done to find the following:

* Incorrect or missing functions
* Interface errors
* Errors on external database access
* Performance error
* Initialization and termination error

##### White Box Testing

This allows the tests to

* Check whether all independent paths within a module have been exercised at least once
* Exercise all logical decisions on their false sides
* Execute all loops and their boundaries and within their boundaries
* Exercise the internal data structure to ensure their validity
* Ensure whether all possible validity checks and validity lookups have been provided to validate data entry.

##### Acceptance Testing

This is the final stage of testing process before the system is accepted for operational use.

The system is tested within the data supplied from the system procurer rather than simulated data

##### Unit Test Case 1

|  |  |
| --- | --- |
| Sl # Test Case | UTC- 1 |
| Name of Test | Pre-processing the image. |
| Expected Result | Input RGB image. Convert to greyscale.  Apply gaussian/weighted filter. |
| Actual output | Same as expected. |
| Remarks | Successful |

**Unit Test Case 2**

|  |  |
| --- | --- |
| Sl # Test Case | UTC- 2 |
| Name of Test | Segmentation. |
| Expected Result | Filtered image as input. Grey to binary image.  Image morphological operation. Removing noise.  Binary image properties. Mask of segment image.  Segmented image. |
| Actual output | Same as expected. |
| Remarks | Successful |

##### Unit Test Case 3

|  |  |
| --- | --- |
| Sl # Test Case | UTC- 3 |
| Name of Test | Feature extraction. |
| Expected Result | Segmented image.  GLCM Feature extraction.  Feature matrix (GLCM fit with labels). |
| Actual output | Same as expected. |
| Remarks | Successful |

**Unit Test Case 4**

|  |  |
| --- | --- |
| Sl # Test Case | UTC- 4 |
| Name of Test | Training. |
| Expected Result | Train ML model with feature matrix.  Saved trained model with pickle. |
| Actual output | Same as expected. |
| Remarks | Successful |

##### Unit Test Case 5

|  |  |
| --- | --- |
| Sl # Test Case | UTC- 5 |
| Name of Test | Classification. |

|  |  |
| --- | --- |
| Expected Result | Input rgb image   1. Pre-processing 2. Segmentation 3. Feature extraction   Pass features to trained ml model |
| Actual output | Same as expected. |
| Remarks | Successful |

**Unit Test Case 6**

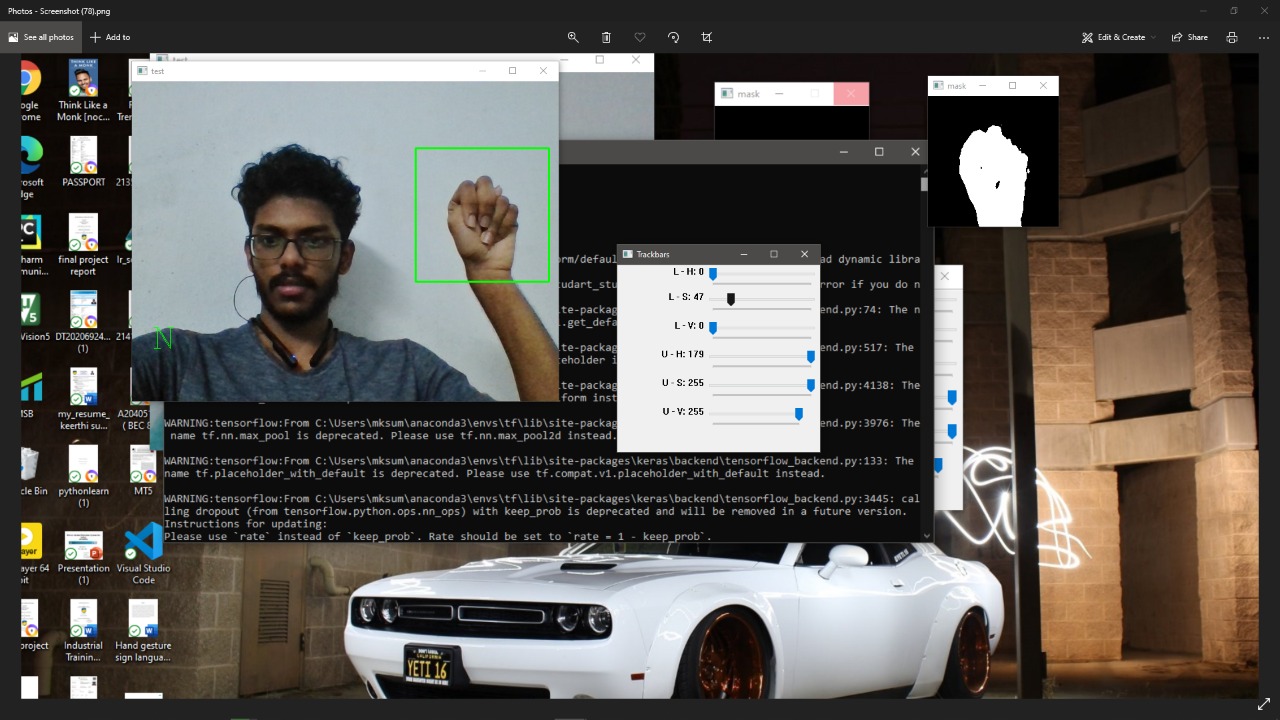
|  |  |
| --- | --- |
| Sl # Test Case | UTC- 5 |
| Name of Test | Sign Recognition. |
| Expected Result | In **Results analysis** are real time detect the sign  and sign recognize when live camera is start after the process in display the result. |
| Actual output | Same as expected. |
| Remarks | Successful |

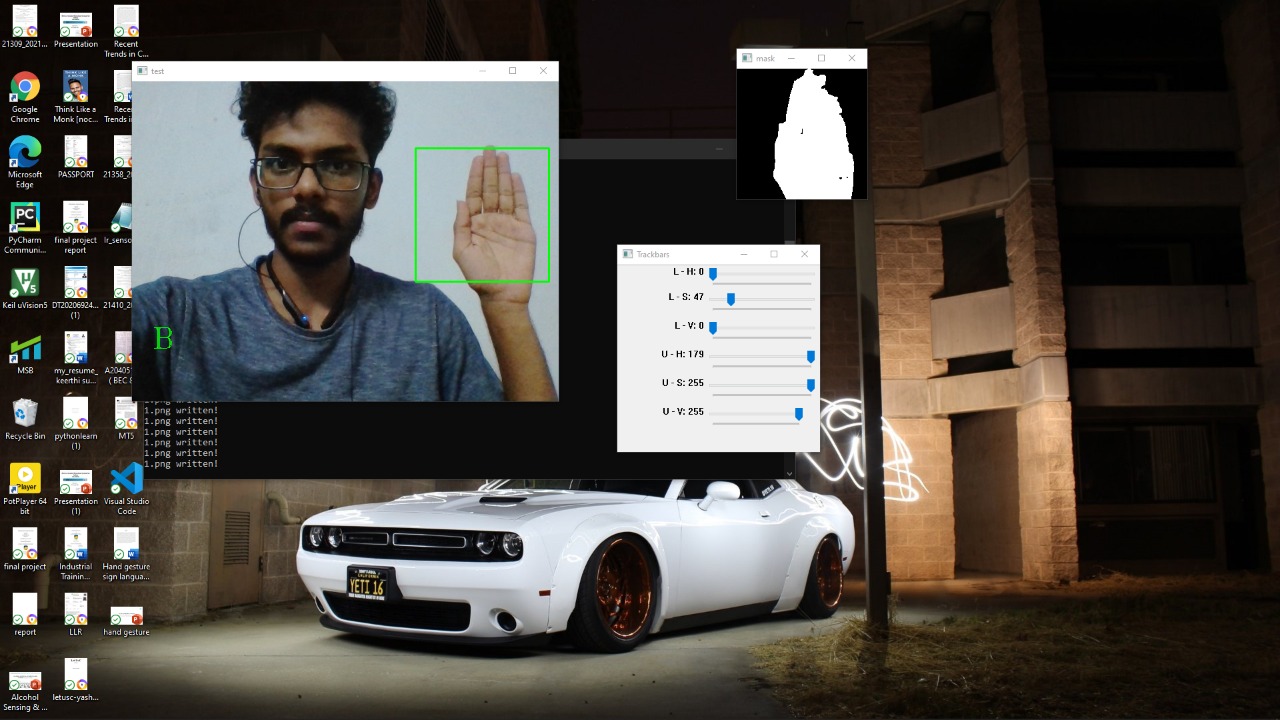
#### Chapter 7

#### 7.1 OUTPUT

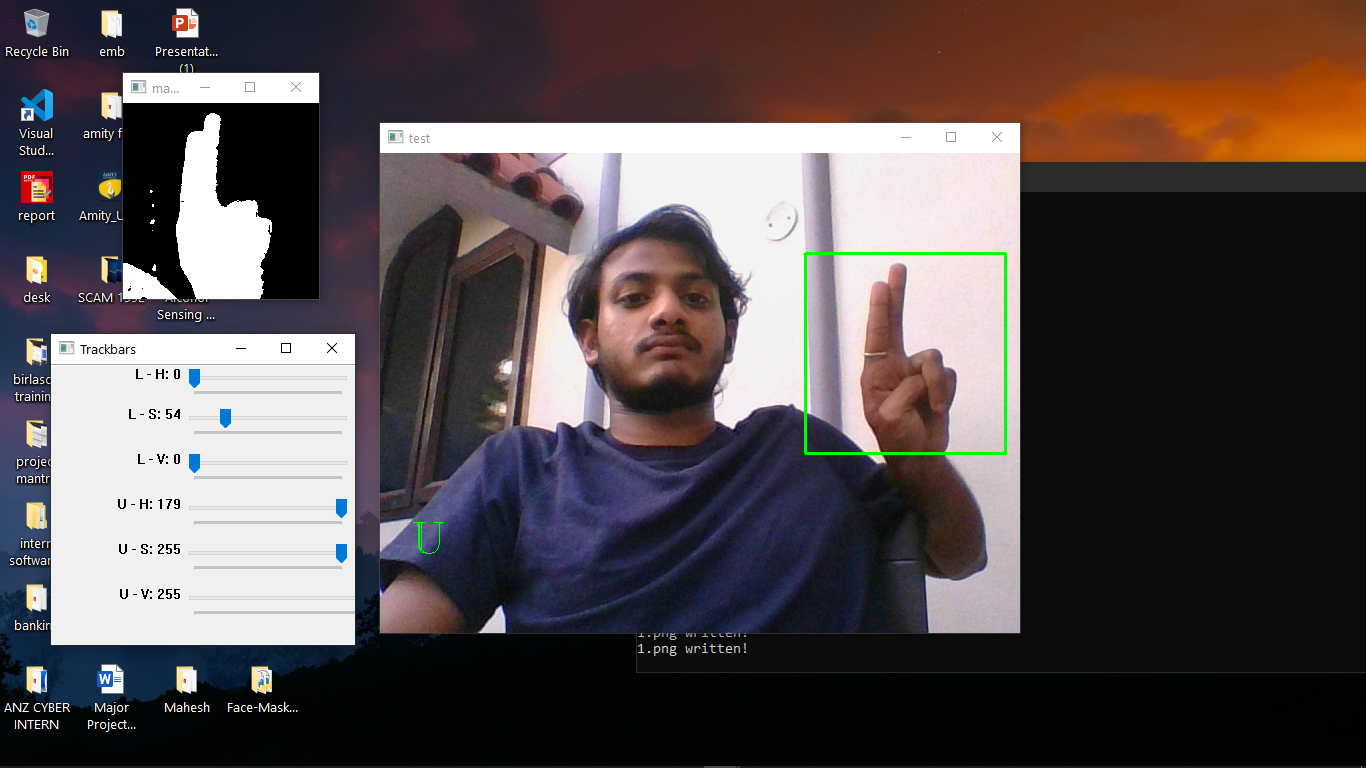
#### 

**Fig 8: Output screen and input image**

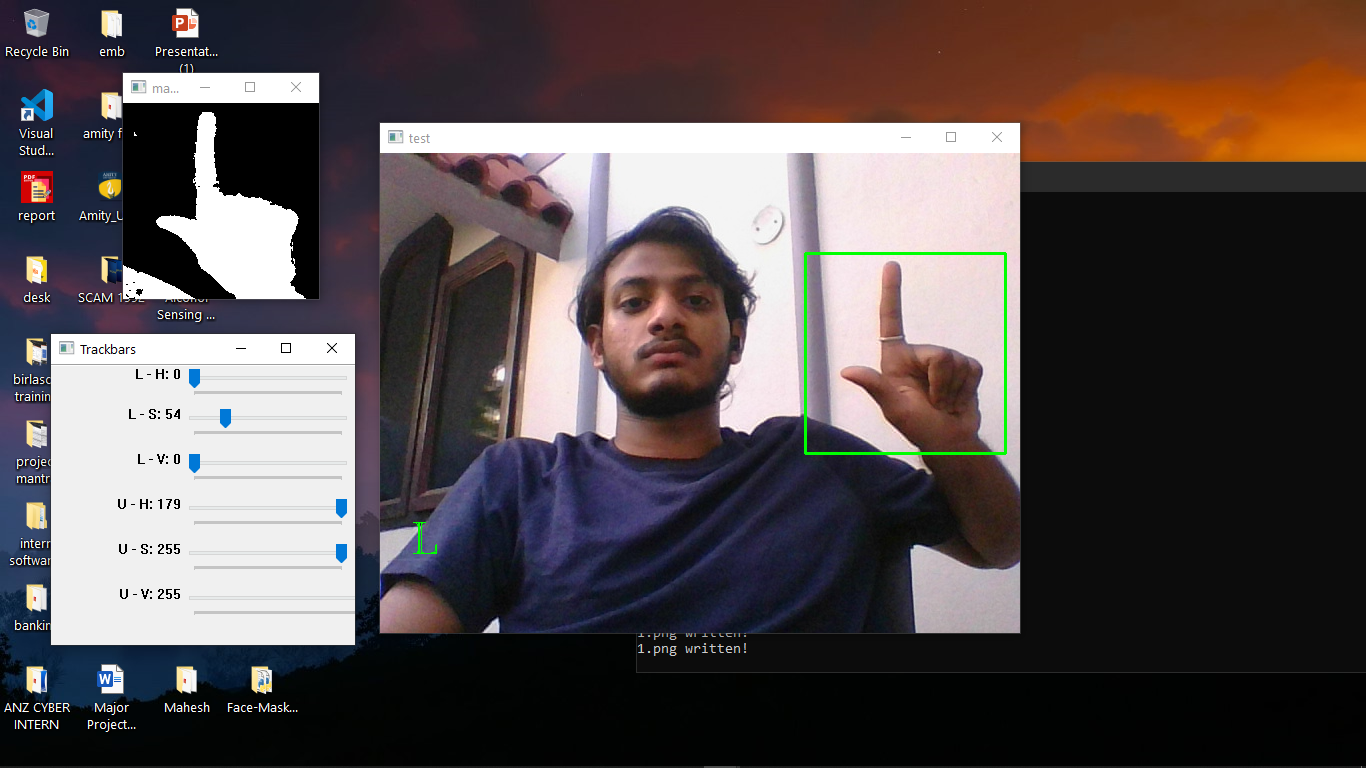
****



##### Fig 9: Input image and Output screen

****

**Fig 10: Output screen and input image**



**Fig 11: Output screen and input image**

##### 7.2 CONCLUSION

In this paper, we have proposed a system for recognizing a dynamic hand words gesture of Indian sign language and conversion of recognized gesture into text and speech and vice versa i.e., dual way communication. In this system skin color filtering technique has used for segmentation. Eigen vectors and Eigen values technique has used for feature extraction. For classification, Eigen value weighted Euclidean Distance based classifier has used. Prediction of words sign using one or both hands, working with Indian Sign language dynamic hand gesture words dataset and dual way communication has proposed in this system.

##### 7.3 FUTURE WORK

* For future work, there’s still so many possibilities of improvement, like noise reduction, dataset collection, better feature extraction, or better model to be used.
* In future, we are looking at developing a system for Indian sign language words that works in real-time. And, we will make efforts to extend our work towards more words and sentences of Indian sign language.

**REFERENCES**

1. A. Chaudhary, J.L. Raheja, K. Das, S. Raheja, "A Survey on Hand Gesture Recognition in context of Soft Computing", Published as Book Chapter in “Advanced Computing” CCIS, Springer Berlin Heidelberg, Vol. 133, 2011, pp. 46-55
2. A. Chaudhary, J.L. Raheja, K. Das, “A Vision based Real Time System to Control Remote Robotic Hand Fingers”, In proceedings of the IEEE International Conference on Computer Control and Automation, South Korea, 1-3 May, 2011, pp. 118-122
3. J.L. Raheja, A. Chaudhary, S. Maheshwari, “Automatic Gesture Pointing Location Detection”, Optik: International Journal for Light and Electron Optics, Elsevier, Vol. 125, Issue 3, 2014, pp. 993-996.
4. A. Chaudhary, K. Vatwani, T. Agrawal, J.L. Raheja, “A Vision-Based Method to Find Fingertips in a Closed Hand”, Journal of Information Processing Systems, Korea, Vol. 8, No. 3, Sep 2012, pp. 399-408.
5. A. Chaudhary, J.L. Raheja, K. Das, S. Raheja, “Fingers' Angle Calculation using Level-Set Method”, Published as Book Chapter in “Computer Vision and Image Processing in Intelligent Systems and Multimedia Technologies”, IGI,

USA, April 2014, pp.191-202

1. D. Sturman, D. Zeltzer, “A survey of glove-based input”, IEEE Transactions on Computer Graphics and Applications, Vol. 14, No. 1, Jan. 1994, pp. 30-39.
2. T.S. Huang, and V.I. Pavlovic, “Hand gesture modeling, analysis and synthesis”, Proceedings of international workshop on automatic face and gesture recognition, 1995, pp.73-79.
3. O.Y. Cho, and et al. “A hand gesture recognition system for interactive virtual environment”, IEEK, 36-s (4), 1999, pp. 70-82.
4. M. Sonka, V. Hlavac, R. Boyle, "Image processing, analysis, and machine vision"2014, Cengage Learning.
5. A. Chaudhary, M. B. L. Manasa, J.L. Raheja, “Light invariant Neuro-Vision System for Elder/Sick people to express their needs into Lingual Description”, Microsoft Research India's annual research symposium TechVista, India, 20 Jan, 2012.
6. A. Chaudhary, J.L. Raheja, "ABHIVYAKTI: A Vision Based Intelligent System for Elder and Sick Persons”, In proceedings of the 3rd IEEE International Conference on Machine Vision, Hong Kong, 28-30 Dec, 2010, pp. 361-364.
7. J L Raheja, Dhiraj, D Gopinath, A. Chaudhary, “GUI System for Elder/Patients in Intensive Care”, In proceedings of the 4th IEEE International Conference on International Technology Management Conference, Chicago, IL USA, 12-15 June, 2014, pp.1-5.
8. J. Rekha, J. Bhattacharya and S. Majumder, "Hand Gesture Recognition for Sign Language: A New Hybrid Approach", 15th IPCV'11, July, 2011, Nevada, USA.
9. A.W. Fitzgibbon, and J. Lockton, “Hand Gesture Recognition Using Computer Vision”, BSc. Graduation Project, Oxford University.
10. <http://en.wikipedia.org/wiki/Principal_component_analysis>.
11. Hand Gesture to Speech Conversion using Matlab,Prof. R RItkarkar and Dr.Anil V Nandi,4th ICCCNT 2013 July 4-6, 2013, Tiruchengode, India.
12. Sign Language to Speech Conversion,Aarthi M and Vijayalakshmi P,2016 Fifth international conference on recent trends in Information technology.
13. Sign Language Recognition System to Aid Deaf-dumb People Using PCA, Arthi M and Vijayalakshmi P, International Journal of Computer Science & Engineering Technology (IJCSET).
14. Full duplex communication system for deaf & dumb people, Shraddha R Ghorpade and Surendra K Waghamare, International Journal of Emerging Technology and Advanced Engineering.