

SIGN LANGUAGE HAND GESTURE DETECTION USING MACHINE LEARNING ALGORITHM

A PROJECT REPORT

***Submitted by***

**RAGHUL.P.V [REGISTER NO: 211417104207] RAJESH KUMARAN.A [REGISTER NO:211414104214]**

**RANJITH.T [REGISTER NO: 211414104222]**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

**IN**

COMPUTER SCIENCE ANDENGINEERING

**PANIMALAR ENGINEERING COLLEGE,CHENNAI-600123.**

**ANNA UNIVERSITY: CHENNAI 600 025**

**APRIL 2021**

**BONAFIDE CERTIFICATE**

Certified that this project report **“HAND GESTURE RECOGNITION USING**

**MACHINE LEARNING”** is the bonafide work of “**…**

**RANJITH .T (REG.NO: 211417104222)**

**RAJESH KUMARAN. A (REG.NO: 211417104214)**

**RAGHUL .P.V (REGNO: 21141710207)”**

Who carried out the project work under my supervision.

SIGNATURESIGNATURE

Dr.S.MURUGAVALLI,M.E.,Ph.D., DR.N.PUGHAZENDI

**HEAD OFTHEDEPARTMENT SUPERVISOR**

**ASSOCIATE PROFESSOR**

DEPARTMENTOFCSE, DEPARTMENT OFCSE,

PANIMALARENGINEERINGCOLLEGE, PANIMALAR ENGINEERINGCOLLEGE, NAZARATHPETTAI, NAZARATHPETTAI,

POONAMALLEE, POONAMALLEE,

CHENNAI-600 123. CHENNAI-600 123

Certifiedthattheabovecandidate(s)was/were examined intheAnnaUniversityProject Viva-Voce Examination held on...........................

## INTERNALEXAMINER EXTERNALEXAMINER

**ACKNOWLEDGEMENT**

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like express our heartfelt and sincere thanks to our Directors

Tmt.C.VIJAYARAJESWARI, Dr.C.SAKTHIKUMAR, M.E., Ph.D., and

**Tmt. SARANYASREE SAKTHIKUMAR B.E,M.B.A,**for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. S. MURUGAVALLI, M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my **Project Guide** …**JAINULABUDEEN.S.A.K**… and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

NAME OF THE STUDENT

RAGHUL PV

RAJESH KUMARAN A

RANJITH T

**ABSTRACT**

Sign Language and gestures is an emerging topic in today’s technologies. The main focus of this is to recognize the human gestures using mathematical algorithms for human computer interaction. With the help of computers, we can communicate with the people who are having hearing impairment. Each of these communicating software has their own limitations when it comes to adapting more versatile hardware in computers. Gesture recognition is one of the essential techniques to build user-friendly interfaces. Usually gestures can be originated from any bodily motion or state, but commonly originate from the face or hand. Gesture recognition enables users to interact with the devices by identifying the actions performed by the user. This project describes how hand gestures are trained to perform certain actions to reflect the actions of the output given by the user. Gestures are used to give the input and the output is calculated.

TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** | iii |
|  | **LIST OF TABLES** | vii |
|  | **LIST OF FIGURES** | viii |
|  | **LIST OF SYMBOLS, ABBREVIATIONS** | ix |
| **1.** | **INTRODUCTION** |  |
|  | 1.1 Overview |  |
|  | 1.2 ProblemDefinition |  |
| **2.** | **LITERATURE SURVEY** |  |
| **3.** | **SYSTEM ANALYSIS** |  |
|  | 3.1. ExistingSystem |  |
|  | 3.2 Proposed System |  |
|  | 3.3 Feasibility study |  |
|  | 3.3.1 Tech feasibility |  |
|  | 3.4 Hardware Requirements  3.5 Software Requirements  3.6 Software Specifications |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
| **4.** | 3.6.1 Machine Learning  3.6.2 Supervised Learning  3.6.3 Classification  3.6.4 Regression  3.6.5 Unsupervised Learning  3.6.5.1 Clustering  3.6.5.2 Clustering Methods  **SYSTEM DESIGN**  4.1 System Architecture  4.2 Usecase Diagram  4.3 Activity Diagram  4.4 Sequence Diagram |  |
| **5.** | **SYSTEM MODULE** |  |
|  | 5.1 Module |  |
| 5.2 Module Description | | |
| **6.** | **SYSTEM IMPLEMENTATION** |  |
|  | 6.1 Client-side coding |  |
|  | 6.2 Server-side coding |  |
|  |  |  |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
| **7.** | **SYSTEM TESTING** |  |
|  | 7.1 Testing Techniques |  |
|  | 7.2 White Box Testing |  |
|  | 7.3 Black Box Testing  7.4 Software Testing Strategies |  |
| **8.**  **9.** | **SCREENSHOT AND OUTPUTS**  **CONCLUSION AND FUTURE REFERENCES** |  |

**10. REFERENCES**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **FIGURE NAME** | **PAGE NO.** |
| 1 | GENERALIZED CLASSIFICATION OF BLOCK DIAGRAM |  |
| 2 | LINEAR REGRESSION MODEL |  |
| 3 | TYPES OF REGRESSION MODEL | 12 |
| 4 | SINGLE GROUP CLUSTERING | 13 |
| 5 | GRID BASED CLUSTERING | 13 |
| 6 | SYSTEM ARCHITECTURE | 45 |
| 7 | USECASE DIAGRAM | 47 |
| 8 | SEQUENCE DIAGRAM | 47 |
| 9 | DATASET IMAGES | 48 |
| 10 | DATASET IMAGES | 48 |
| 11 | TRAINING MODEL | 49 |
| 12 | TRAINING GESTURE | 50 |
| 13 | ENGLISH WEBUI | 50 |
| 14 | FRENCH WEBUI | 51 |
| 15 | HINDI WEBUI | 51 |
|  |  |  |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **ABBREVIATION** | **DESCRIPTION** |
| LQE | Linear Quadratic Estimation |
| HSB | Hue Saturation Brightness |
| OCR | Optical Character Recognition |
| SGD | Stochastic Gradient Descent |
| STING | Statistical Information Grid |
| CLIQUE | Clustering in Quest |
| DBSCAN | Density Based Spatial Clustering Of Application With Noise |

**CHAPTER 1**

**INTRODUCTION**

Humans are able to recognize body and sign language easily. This is possible due to the combination of vision and synaptic interactions that were formed along brain development. The requirements for hand detection involve the input image from the webcam. The image should be fetched with a speed of 20 frames per second. Distance should also be maintained between the hand and the camera. Approximate distance that should be between hands the camera is around 30 to 100 cm. The video input is stored frame by frame into a matrix after preprocessing. In order to replicate this skill in computers, some problems need to be solved: how to separate objects of interest in images and which image capture technology and classification technique are more appropriate, among others. In this way, these devices are capable of capturing human gestures, developing a new medium of human-machine interaction. The uses of these devices are present in the most diverse areas, such as robotics, medicine, sign language translation, computer graphics, and augmented reality. Gesture recognition methodologies are usually divided into two categories: static or dynamic. Static gestures are those that only require the processing of a single image at the input of the classifier, the advantage of this approach is the lower computational cost. Dynamic gestures require the processing of image sequences and more complex gesture recognition approaches. In the literature, we can find several recognition methodologies based on supervised and unsupervised learning. The method of neural networks to classify the data extracted from the images which a special type of neural network is used, called learning vector quantization. Convolutional neural networks and stacked de noising auto encoder. In this work, we used two image bases of 24 gestures.

Some segmentation techniques and the use of convolutional neural network (CNNs) for classification. Thus, with the proposed methodology, we demonstrated that with simple architectures of convolutional neural networks, it is possible to achieve excellent results for static gesture classification. The final sections of this work show the results we obtained a discussion and comparison with other works and, lastly, our conclusions and perspectives for future work.

Robots are used successfully in many areas today, particularly in industrial production, military operations, deep sea drilling, and space exploration. This success drives the interest in the feasibility of using robots in human social environments, particularly in the care of the aged and the handicapped. In social environments, humans communicate easily and naturally by both speech (audio) and gesture (vision) without the use of any external devices (like keyboards) requiring special training. Robots have to adapt to human modes of communication to promote a more natural interaction with humans. Given a choice between speech and gesture, some researchers have opined that gesture recognition would be more reliable than speech recognition because the latter would need a greater number of training datasets to deal with the greater variability in human voice and speech.

**1.1 OVERVIEW**

This project is about implementing the control of a robot through simple hand gestures. The main motivation is the desirability of developing robots that can interact smoothly with humans without the need of any special devices. The objectives of the project are: 1) Study and apply the needed tools, namely: a) An omnidirectional robot from the Massey University Robot Laboratory b) The Open CV Computer Vision Library (version 2.0) c) Algorithms for computer vision and artificial intelligence 2) Develop a computer vision application for simple gesture recognition 3) Test the computer application 4) Document the results of the project During the project, four gestures were chosen to represent four navigational commands for the robot, namely Move Forward, Move Left, Move Right, and Stop. A simple computer vision application was written for the detection and recognition of the four gestures and their translation into the corresponding commands for the robot.

The appropriate Open CV functions and image processing algorithms for the detection and interpretation of the gestures were used. Thereafter, the program was tested on a webcam with actual hand gestures in real-time and the results were observed. The results of the project demonstrated that a simple computer vision application can be designed to detect and recognize simple hand gestures for robot navigational control based on simple heuristic rules. The program was able to correctly interpret the gestures and translate it into the corresponding commands most of the time. This report will present a short review of computer vision and digital image processing, introduce Open CV as a popular tool for the development of computer vision applications, discuss pattern recognition and classifiers, Hu's Moment Invariants, and the Ada Boost algorithm, discuss the project tools and methodology, outline the steps undertaken to complete the project, and discuss the results and conclusion.

Computer Vision and Digital Image Processing, The sense of sight is arguably the most important of man's five senses. It provides a huge amount of information about the world that is rich in detail and delivered at the speed of light. However, human vision is not without its limitations, both physical and psychological. Through digital imaging technology and computers, man has transcending many visual limitations. He can see into far galaxies, the microscopic world, the sub-atomic world, and even “observe” infra-red, x-ray, ultraviolet and other spectra for medical diagnosis, meteorology, surveillance, and military uses, all with great success. While computers have been central to this success, for the most part man is the sole interpreter of all the digital data. For a long time, the central question has been whether computers can be designed to analyze and acquire information from images autonomously in the same natural way humans can. According to Gonzales and Woods , this is the province of computer vision, which is that branch of artificial intelligence that ultimately aims to “use computers to emulate human vision, including learning and being able to make inferences and taking actions based on visual inputs.” The main difficulty for computer vision as a relatively young discipline is the current lack of a final scientific paradigm or model for human intelligence and human vision itself on which to build the infrastructure for computer or machine learning. The use of images has an obvious drawback. Humans perceive the world in 3D, but current visual sensors like cameras capture the world in 2D images.

The result is the natural loss of a good deal of information in the captured images. Without a proper paradigm to explain the mystery of human vision and perception, the recovery of lost information (reconstruction of the world) from 2D images represents a difficult hurdle for machine vision. However, despite this limitation, computer vision has progressed, riding mainly on the remarkable advancement of decades-old digital image processing techniques, using the science and methods contributed by other disciplines such as optics, neurobiology, psychology, physics, mathematics, electronics, computer science, artificial intelligence and others. Computer vision techniques and digital image processing methods both draw the proverbial water from the same pool, which is the digital image, and therefore necessarily overlap. Image processing takes a digital image and subjects it to processes, such as noise reduction, detail enhancement, or filtering, for the purpose of producing another desired image as the end result. For example, the blurred image of a car registration plate might be enhanced by imaging techniques to produce a clear photo of the same so the police might identify the owner of the car. On the other hand, computer vision takes a digital image and subjects it to the same digital imaging techniques but for the purpose of analyzing and understanding what the image depicts. For example, the image of a building can be fed to a computer and thereafter be identified by the computer as a residential house, a stadium, high-rise office tower, shopping mall, or a farm barn. Russell and NORVIG identified three broad approaches used in computer vision to distill useful information from the raw data provided by images. The first is the feature extraction approach, which focuses on simple computations applied directly to digital images to measure some useable characteristic, such as size. This relies on generally known image processing algorithms for noise reduction, filtering, object detection, edge detection, texture analysis, computation of optical flow, and segmentation, which techniques are commonly used to pre-process images for subsequent image analysis. This is also considered an “uninformed” approach. The second is the recognition approach, where the focus is on distinguishing and labeling objects based on knowledge of characteristics that sets of similar objects have in common, such as shape or appearance or patterns of elements, sufficient to form classes. Here computer vision uses the techniques of artificial intelligence in knowledge representation to enable a “classifier” to match classes to objects based on the pattern of their features or structural descriptions. A classifier has to “learn” the patterns by being fed a training set of objects and their classes and achieving the goal of minimizing mistakes and maximizing successes through a step-by-step process of improvement.

**1.2 PROBLEM DEFINITION**

In this project, with the help of CNN, we are going to calculate the command that we are inputting to the system by sign gestures and the gestures will reflect. The probable command which the user is trying to reflect is shown in the output bar. With the sign language user is showing, the computer recognizes the input and the command which the user is intended is shown as an output. This can help to interact people as a medium who are want to talk with people who are communicating with people who knew the gestures of American Sign Language and the people who aren’t with the help of our Web UI.

**CHAPTER 2**

**LITERATURE SURVEY**

**1. Hand Gesture Recognition for Human Computer Interaction, International Conference on Image Information Processing India**

A hand gesture recognition system provides a natural, innovative and modern way of nonverbal communication. It has a wide area of application in human computer interaction and sign language. The intention of this paper is to discuss a novel approach of hand gesture recognition based on detection of some shape based features. The setup consists of a single camera to capture the gesture formed by the user and take this hand image as an input to the proposed algorithm. The overall algorithm divided into four main steps, which includes segmentation, orientation detection, feature extraction and classification. The proposed algorithm is independent of user characteristics. It does not require any kind of training of sample data. The proposed Implemented algorithm has been tested on 390 images, gives a recognition rate of approximately 92% and average elapsed time of 2.76 sec. It takes a less computation time as compare to other approaches.

**2. Parkinson’s Comparative Study of Hand Gesture Recognition System, AIRCC Digital Library – 2012.**

Human imitation for his surrounding environment makes him interfere in every details of this great environment, hear impaired people are gesturing with each other for delivering a specific message, this method of communication also attracts human imitation attention to cast it on human-computer interaction. The faculty of vision based gesture recognition to be a natural, powerful, and friendly tool for supporting efficient interaction between human and machine. In this paper a review of recent hand gesture recognition systems is presented with description of hand gestures modeling analysis and recognition. A comparative study included in this paper with focusing on different segmentation, features extraction and recognition tools, research advantages and drawbacks are provided as well.

**3 Hand Gesture Recognition Systems: A Survey, International Journal of Computer Applications Volume 71– No.15, May 2013.**

Gesture was the first mode of communication for the primitive cave men. Later on human civilization has developed the verbal communication very well. But still nonverbal communication has not lost its weightage. Such non – verbal communication are being used not only for the physically challenged people, but also for different applications in diversified areas, such as aviation, surveying, music direction etc. It is the best method to interact with the computer without using other peripheral devices, such as keyboard, mouse. Researchers around the world are actively engaged in development of robust and efficient gesture recognition system, more specially, hand gesture recognition system for various applications. The major steps associated with the hand gesture recognition system are; data acquisition, gesture modeling, feature extraction and hand gesture recognition. There are several sub-steps and methodologies associated with the above steps. Different researchers have followed different algorithm or sometimes have devised their own algorithm. The current research work reviews the work carried out in last twenty years and a brief comparison has been performed to analyze the difficulties encountered by these systems, as well as the limitation. Finally the desired characteristics of a robust and efficient hand gesture recognition system have been described.

**4.Comparative Study of Hand Gesture Recognition Algorithms, International**

**Journal of Research in Computer and Communication Technology, Volume**

**3, Issue 4, April- 2014.**

This survey paper proposes a real time implementation and novel methods for a hand-pose estimation that can be used for vision-based human interfaces. In this paper different methodologies, segmentation, feature extraction, classifiers are discussed and compared. The experimental results indicate the feasibility of different methodologies for vision-based interfaces in different environmental conditions although the methodologies discussed for various database and specified classifiers were used for faster implementation for real-time processing.

**5. Navigation of PowerPoint Using Hand Gestures, International Journal of Science and Research (IJSR) 2015**

Computer applications require interaction between human and computer. This interaction needs to be unrestricted and it made challengeable accounts as compared to traditional input devices such as keyboard, mouse, pen etc. Hand gestures are used in daily life while communication. They are more natural in interaction, compared with those devices mentioned above. Human computer interaction becomes easy, with use of hand as a device. Use of hand gestures to operate machine would make interaction interesting. Gesture recognition has gained a lot of importance. Hand gestures are used to control various applications like VLC media player, robot control, gaming, virtual mouse etc. Use of gesture makes interaction easy, convenient and does not require any extra device. Vision and audio recognition can be used together. But audio commands may not work in noisy environment. The presented developed system uses hand gestures to control power point navigation. Here both static and dynamic gestures are used together for navigation. Use of gestures would make the presenter to handle the presentation more easily.

**6. Real time finger tracking and contour detection for gesture recognition using Open CV, IEEE Conference May 2015, Pune India.**

Gestures are important for communicating information among the human. Nowadays new technologies of Human Computer Interaction (HCI) are being developed to deliver user's command to the robots. Users can interact with machines through hand, head, facial expressions, voice and touch. The objective of this paper is to use one of the important modes of interaction i.e. hand gestures to control the robot or for offices and household applications. Hand gesture detection algorithms are based on various machine learning methods such as neural networks, support vector machine, and Adaptive Boosting (Ada Boost). Among these methods, Ada Boost based hand-pose detectors are trained with a reduced HAAR like feature set to make the detector robust. The corresponding context-free grammar based proposed method gives effective real time performance with great accuracy and robustness for more than four hand gestures. Rectangles are creating some problem due to that we have also implement the alternate representation method for same gestures i.e. fingertip detection using convex hull algorithm.

**7 A Real-time Hand Gesture Recognition and Human-Computer Interaction System, Research Paper April 2017.**

In this project, we design a real-time human-computer interaction system based on hand gesture. The whole system consists of three components: hand detection, gesture recognition and human-computer interaction (HCI) based on recognition; and realizes the robust control of mouse and keyboard events with a higher accuracy of gesture recognition. Specifically, we use the convolutional neural network (CNN) to recognize gestures and make it attainable to identify relatively complex gestures using only one cheap monocular camera. We introduce the KALMAN filter, or LQE to estimate the hand position based on which the mouse cursor control is realized in a stable and smooth way. During the HCI stage, we develop a simple strategy to avoid the false recognition caused by noises - mostly transient, false gestures, and thus to improve the reliability of interaction. The developed system is highly extendable and can be used in human-robotic or other human-machine interaction scenarios with more complex command formats rather than just mouse and keyboard events.

**8 Detection and Recognition of Gestures To Control The System Applications by Neural Networks**

This implementation focuses on detection of hand gestures using java and neural networks. It is divided into two phases. Detection module using java where in the hand is detected using background subtraction and conversion of video feed into HSB video feed thus detecting skin pixels. The second module is the prediction module; a convolutional neural network is used. The input feed image is gained from Java. The input image is fed into the neural network and is analyzed with respect to the dataset images. One of the limitations of this system is that it requires socket programming in order to connect java and python modules.

**CHAPTER 3**

**SYSTEM ANALYSIS**

* 1. **EXISTING SYSTEM**

Gesture recognition has multiple applications in medical and engineering fields. The problem of hand gesture recognition consists of identifying, at any moment, a given gesture performed by the hand. In this work, we propose a new model for hand gesture recognition in real time. The input of this model is the surface electromyography measured by the commercial sensor the MYO armband placed on the forearm. The output is the label of the gesture executed by the user at any time. The proposed model is based on the Λ-nearest neighbor and dynamic time warping algorithms. This model can learn to recognize any gesture of the hand. To evaluate the performance of our model, we measured and compared its accuracy at recognizing 5 classes of gestures to the accuracy of the proprietary system of the MYO armband. As a result of this evaluation, we determined that our model performs better (86% accurate) than the MYO system (83%).

* 1. **PROPOSED SYSTEM**

The physical movement of the human hand produces gestures, and hand gesture recognition. In this paper, the human hand gestures are detected and recognized using machine learning. This process flow consists of hand region of interest segmentation using mask image, fingers segmentation, normalization of segmented finger image and finger recognition using CNN classifier. The hand region of the image is segmented from the whole image using mask images. The adaptive histogram equalization method is used as enhancement method for improving the contrast of each pixel in an image. In this paper, connected component analysis algorithm is used in order to segment the finger tips from hand image. The segmented finger regions from hand image are given to the CNN classification algorithm which classifies the image into various classes. The proposed hand gesture detection and recognition methodology using CNN classification approach with enhancement technique stated in this paper achieves high performance with state-of-the-art methods.

**3.3 FEASIBILITY STUDY**

A feasibility study is carried out to select the best system that meets performance requirements. The main aim of the feasibility study activity is to determine that it would be financially and technically feasible to develop the product.

**3.3.1 TECHNICAL FEASIBILITY**

This is concerned with specifying the software will successfully satisfy the user requirement. Open source and business-friendly and it is truly cross platform, easily deployed and highly extensible.

**3.3.2 ECONOMIC FEASIBILITY**

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. The enhancement of the existing system doesn’t incur any kind of drastic increase in the expenses. Java script is open source and ready available for all users. Since the project is run in Java script and JUPYTER notebook hence is cost efficient.

**3.4 HARWARE REQUIREMENTS**

Processor : Intel Pentium Dual Core 2.00GHz

Hard disk : 40 GB

RAM : 2 GB (minimum)

**3.5 SOFTWARE REQUIREMENTS**

* JUPYTERNotebook
* Javascript
* Tensorflow
* Teachable Machine
* Glitch UI

**3.6 SOFTWARE SPECIFICATION**

**3.6.1 MACHINE LEARNING**

Machine learning is a subfield of artificial intelligence (AI). The goal of machine learning generally is to understand the structure of data and fit that data into models that can be understood and utilized by people. Although machine learning is a field within computer science, it differs from traditional computational approaches. In traditional computing, algorithms are sets of explicitly programmed instructions used by computers to calculate or problem solve. Machine learning algorithms instead allow for computers to train on data inputs and use statistical analysis in order to output values that fall within a specific range. Because of this, machine learning facilitates computers in building models from sample data in order to automate decision-making processes based on data inputs.

Any technology user today has benefitted from machine learning. Facial recognition technology allows social media platforms to help users tag and share photos of friends. OCR technology converts images of text into movable type. Recommendation engines, powered by machine learning, suggest what movies or television shows to watch next based on user preferences. Self-driving cars that rely on machine learning to navigate may soon be available to consumers. Machine learning is a continuously developing field. Because of this, there are some considerations to keep in mind as you work with machine learning methodologies, or analyze the impact of machine learning processes.

Here in this thesis, we are providing basic info of the common machine learning methods of supervised and unsupervised learning, and common algorithmic approaches in machine learning, including the k-nearest neighbor algorithm, decision tree learning, and deep learning.

**3.6.2 SUPERVISED LEARNING**

In machine learning, tasks are generally classified into broad categories. These categories are based on how learning is received or how feedback on the learning is given to the system developed. Two of the most widely adopted machine learning methods are supervised learning which trains algorithms based on example input and output data that is labeled by humans, and unsupervised learning which provides the algorithm with no labeled data in order to allow it to find structure within its input data. Let’s explore these methods in more detail.

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. Techniques of Supervised Machine Learning algorithms include linear and logistic regression, multi-class classification, Decision Trees and support vector machines. Supervised learning requires that the data used to train the algorithm is already labeled with correct answers. For example, a classification algorithm will learn to identify animals after being trained on a dataset of images that are properly labeled with the species of the animal and some identifying characteristics. Supervised learning problems can be further grouped into Regression and Classification problems. Both problems have as goal the construction of a succinct model that can predict the value of the dependent attribute from the attribute variables. The difference between the two tasks is the fact that the dependent attribute is numerical for regression and categorical for classification.

**3.6.3 CLASSIFICATION**

As the name suggests, Classification is the task of “classifying things” into sub- categories, but by a machine. If that doesn’t sound like much, imagine your computer being able to differentiate between you and a stranger between potato and tomato. Between an A grade and F In Machine Learning and Statistics, Classification is the problem of identifying to which of a set of categories (sub populations), a new observation belongs to, on the basis of a training set of data containing observations and whose categories membership is known.

**TYPES OF CLASSIFICATION**

Classification is of two types:

• **Binary Classification:** When we have to categorize given data into 2 distinct classes. Example – On the basis of given health conditions of a person, we have to determine whether the person has a certain disease or not.

• **Multiclass Classification:** The number of classes is more than 2. For Example

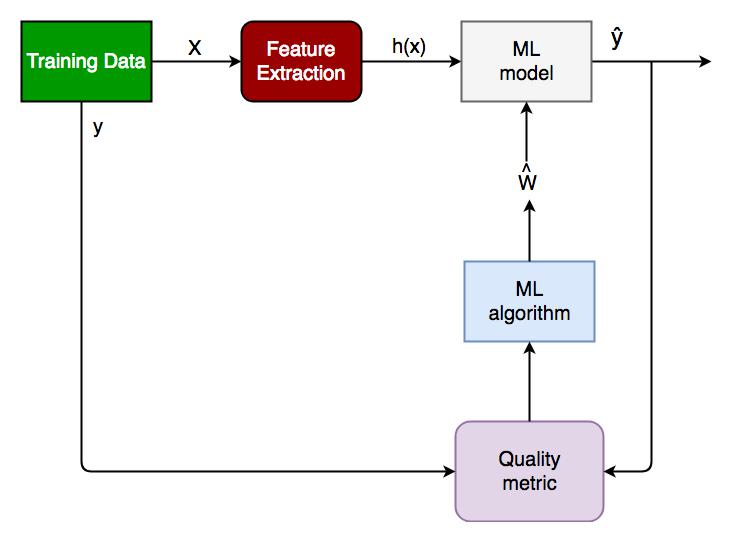
– On the basis of data about different species of flowers, we have to determine which species does our observation Belongs.

Fig 2: Binary and Multiclass Classification. Here x1 and x2 are our variables upon which the class is predicted. Suppose we have to predict whether a given patient has a certain disease or not, on the basis of 3 variables, called features. It means there are two possible Outcomes:

1. The patient has the said disease. Basically a result labeled “Yes” or “True”.

2. The patient is disease free. A result labeled “No” or “False”.

This is a binary classification problem. We have a set of observations called training data set, which comprises of sample data with actual classification results. We train a model, called Classifier on this data set, and use that model to predict whether a certain patient will have the



**Fig 1: Generalized Classification Block Diagram.**

1. X: pre-classified data, in the form of a N\*M matrix. N is the no. of observations and M is the number of features

2. Y: An N-d vector corresponding to predicted classes for each of the N observations.

3. Feature Extraction: Extracting valuable information from input X using a series of transforms.

4. ML Model: The “Classifier” we’ll train.

5. Y’: Labels predicted by the Classifier.

6. Quality Metric: Metric used for measuring the performance of the model.

7. ML algorithm: The algorithm that is used to update weights which update the model and learns iteratively.

Types of Classifiers (Algorithms)

There are various types of classifiers. Some of them are:

• Linear classifier: Logistic Regression

• Tree Based Classifiers: Decision Tree Classifier

• Support Vector Machines

• Artificial Neural Networks

• Bayesian Regression

• Gaussian Naive Bayes Classifiers

• SGD Classifier

• Ensemble Methods: Random Forests, Ada Boost, Bagging Classifier, Voting Classifier, Extra Trees Classifier

Practical Applications of Classification

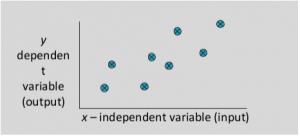
• Google’s Self driving car uses deep learning enabled classification techniques which enables it to detect and classify obstacles.

• Spam E-mail filtering is one of the most widespread and well recognized uses of Classification techniques.

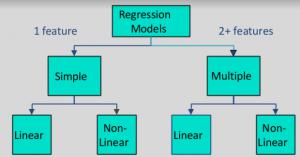
• Detecting Health Problems, Facial Recognition, Speech Recognition, Object Detection, Sentiment Analysis all use Classification at their core.

**3.6.4 REGRESSION**

A regression problem is when the output variable is a real or continuous value, such as “salary” or “weight”. Many different models can be used the simplest is the linear regression. It tries to fit data with the best hyper-plane which goes through the points.



**Fig 2 : Linear Regression Mode**

****

**Fig 3: Types of Regression Models**

**3.6.5 UNSUPERVISED LEARNING**

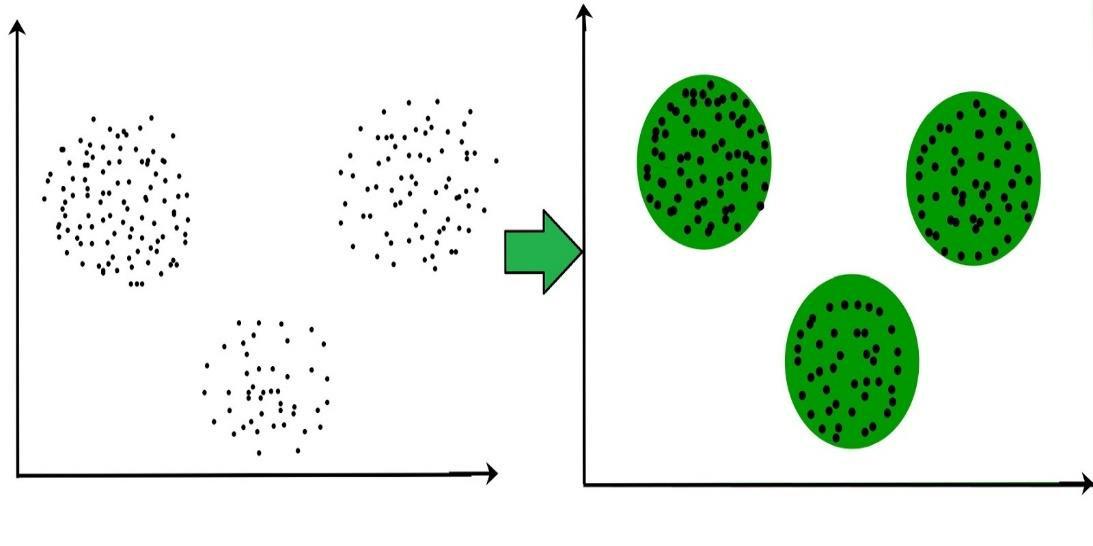
Unsupervised learning is where we only have input data (X) and no corresponding output variables. The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data. These are called unsupervised learning because unlike supervised learning above there is no correct Answer and there is no teacher. Algorithms are left to their own devises to discover and present the interesting structure in the data. Unsupervised learning problems can be further grouped into clustering and association problems

**3.6.5.1 CLUSTERING**

It is basically a type of unsupervised learning method. An unsupervised learning method is a method in which we draw references from datasets consisting of input data without labeled responses. Generally, it is used as a process to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples. Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.

For example, the data points in the graph below clustered together can be classified into one single group. We can distinguish the clusters, and we can identify that there are 3 clusters in the below picture.

These data points are clustered by using the basic concept that the data point lies within the given constraint from the cluster center. Various distance methods and techniques are used for calculation of the outliers.



**Fig 4: Single group clustering**

Clustering is very much important as it determines the intrinsic grouping among the unlabeled data present. There are no criteria for a good clustering. It depends on the user, what is the criteria they may use which satisfy their need. For instance, we could be interested in finding representatives for homogeneous groups (data reduction), in finding “natural clusters” and describe their unknown properties (“natural” data types), in finding useful and suitable groupings (“useful” data classes) or in finding unusual data objects (outlier detection). This algorithm must make some assumptions which constitute the similarity of points and each assumption make different and equally valid clusters.

**3.6.5.1.1 Clustering Method:**

1. **Density-Based Methods:** These methods consider the clusters as the dense region having some similarity and different from the lower dense region of the space. These methods have good accuracy and ability to merge two clusters. Example DBSCAN

(Density-Based Spatial Clustering of Applications with Noise) , OPTICS (Ordering Points to Identify Clustering Structure) etc.

2. **Hierarchical Based Methods:** The clusters formed in this method forms a tree type structure based on the hierarchy. New clusters are formed using the previously formed one. It is divided into two categories.

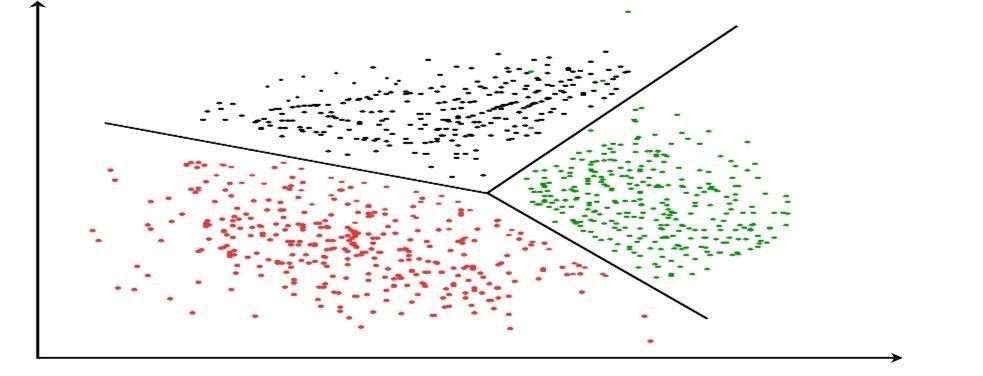
• Agglomerative (bottom up approach)

• Divisive (top down approach)

3. **Partitioning Methods:** These methods partition the objects into k clusters and each partition forms one cluster. This method is used to optimize an objective criterion similarity function such as when the distance is a major parameter example K-means, CLARANS (Clustering Large Applications based upon randomized Search) etc.

4. **Grid-based Methods:** In this method the data space are formulated into a finite number of cells that form a grid-like structure. All the clustering operation done on these grids are fast and independent of the number of data objects example STING (Statistical Information Grid), wave cluster, CLIQUE (clustering In Quest) etc.

Clustering Algorithms:

* K-Means Clustering.
* Mean-Shift Clustering for a single sliding window.
* The entire process of Mean-Shift Clustering.
* DBSCAN Smiley Face Clustering.
* EM Clustering using GMMS.
* Agglomerative Hierarchical Clustering.

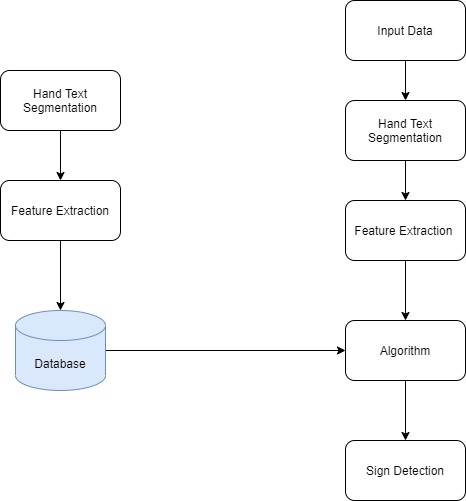
**Fig 5: Grid based Clustering**

**CHAPTER 4**

**SYSTEM DESIGN**

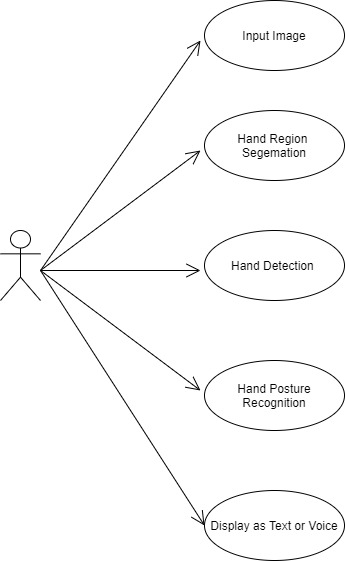
**4.1 SYSTEM ARCHITECTURE**

System architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal encryption and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.



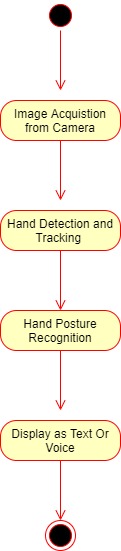
**Fig 6: System Architecture**

**4.2 USECASE DIAGRAM**

****

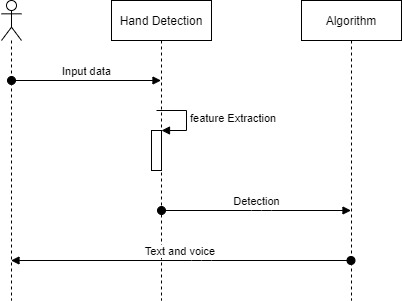
**Fig 7: Use Case Diagram**

**4.3 ACTIVITY DIAGRAM**

****

**Fig 8: Activity Diagram**

**4.4 SEQUENCE DIAGRAM**

****

**Fig 9: Sequence Diagram**

**CHAPTER 5**

**SYSTEM MODULE**

**5.1 MODULE**

* Capture scene
* Hand Detection
* Recognition
* Working

**5.2 MODULE DESCRIPTION**

* **Capture scene**

Capture images through a web camera. It is used as an input to the system that is built. Preprocessing: Images that are captured through the webcam are compared with the dataset to recognize the valid hand movements that are needed to perform the required actions.

* **Hand Detection**

The requirements for hand detection involve the input image from the webcam. The image should be fetched with a speed of 20 frames per second. Distance should also be maintained between the hand and the camera. Approximate distance that should be between hand and camera is around 30 to 100 cm. The video input is stored frame by frame into a matrix after preprocessing.

* **Recognition**

This step involves: Gesture Recognition: The number of fingers present in the hand gesture is determined by making use of defect points present in the gesture. The resultant gesture obtained is fed through a 3Dimensional Convolutional Neural Network consecutively to recognize the current gesture. Performing action: The recognized gesture is used as an input to perform the actions required by the user. These actions include zooming in, zooming out and swiping the page left or right.

* **Working**

1. Detection of Hand: This module detects the hand gesture by capturing an image through the web camera.

2. Finger Tip Detection and Defect Calculation: A boundary box is drawn around the hand for which the finger tips are detected and the background space around iis eliminated.

**CHAPTER 6**

**SYSTEM IMPLEMENTATION**

**Client-side Programing:**

<div>Teachable Machine Image Model</div>

<button type="button" onclick="init()">Start</button>

<div id="webcam-container"></div>

<div id="label-container"></div>

<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.3.1/dist/tf.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/@teachablemachine/image@0.8/dist/teachablemachine-image.min.js"></script>

<script type="text/java script">

// More API functions here:

// <https://github.com/googlecreativelab/teachablemachine-> community/tree/master/libraries/image

// the link to your model provided by Teachable Machine export panel

Const URL = "./my model/";

let model, webcam, label Container, max Predictions;

// Load the image model and setup the webcam

async function init() {

constmodelURL = URL + "model.json";

constmetadataURL = URL + "metadata.json";

// load the model and metadata

// Refer to tmImage.loadFromFiles() in the API to support files from a file picker

// or files from your local hard drive

// Note: the pose library adds "tmImage" object to your window (window.tmImage)

model = await tmImage.load(modelURL, metadataURL);

maxPredictions = model.getTotalClasses();

// Convenience function to setup a webcam

const flip = true; // whether to flip the webcam

webcam = new tmImage.Webcam(200, 200, flip); // width, height, flip

awaitwebcam.setup(); // request access to the webcam

awaitwebcam.play();

window.requestAnimationFrame(loop);

// append elements to the DOM

document.getElementById("webcam-container").appendChild(webcam.canvas);

labelContainer = document.getElementById("label-container");

for (let i = 0; i <maxPredictions; i++) { // and class labels

labelContainer.appendChild(document.createElement("div"));

}

}

async function loop() {

webcam.update(); // update the webcam frame

await predict();

window.requestAnimationFrame(loop);

}

// run the webcam image through the image model

async function predict() {

// predict can take in an image, video or canvas html element

const prediction = await model.predict(webcam.canvas);

for (let i = 0; i <maxPredictions; i++) {

constclassPrediction =prediction[i].className + ": " + prediction[i].probability.toFixed(2);

labelContainer.childNodes[i].innerHTML = classPrediction;

}

}

</script>

**Server-side Programing:**

:root {

--box-size: 200px;

--gray: #9aa0a6;

}

body {

font-family: 'Barlow', sans-serif;

letter-spacing: 0.5px;

display: flex;

flex-direction: column;

align-items: center;

height: 100vh;

justify-content: space-between;

}

section {

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

width:100%;

}

section#model {

margin-top: 90px;

}

section#info {

margin-top: 50px;

}

section > \* {

margin: 14px;

}

a,h1,h2 {

text-align: center;

}

a {

color: #000;

}

h1 {

font-size: 24px;

line-height: 29px;

}

h2 {

font-size: 18px;

line-height: 22px;

}

#webcam-wrapper {

width: var(--box-size);

height: var(--box-size);

background-color: var(--gray);

border-radius: 4px;

overflow: hidden;

position: relative;

}

.loader {

position: absolute;

top:50%;

left:50%;

border: 4px solid #ccc; /\* Light grey \*/

border-top: 4px solid #fff; /\* Blue \*/

border-radius: 50%;

width: 60px;

height: 60px;

animation: spin 2s linear infinite;

}

@keyframes spin {

0% { transform: translate(-50%,-50%) rotate(0deg); }

100% { transform: translate(-50%,-50%) rotate(360deg); }

}

#webcam-wrapper>canvas {

position: relative;

}

#webcam {

width: var(--box-size);

height: var(--box-size);

transform: rotateY(180deg);

}

#graph-wrapper > div {

display: flex;

flex-direction: row;

align-items: center;

justify-content: center;

margin: 10px 0;

}

#graph-wrapper > div > \* {

display: inline-block;

width: var(--box-size);

margin: 0 10px;

}

#graph-wrapper > div > :first-child {

text-align: right;

}

#graph-wrapper > div > progress {

-webkit-appearance: none;

appearance: none;

height: 28px;

}

#graph-wrapper > div > progress::-webkit-progress-bar {

border-radius: 4px;

background-color: var(--color-light);

border: 1px solid var(--color);

}

#graph-wrapper > div > progress::-webkit-progress-value {

border-radius: 2px;

background-color: var(--color);

}

footer

{

width: 100%;

text-align: right;

padding: 25px;

}

footer \*

{

color: var(--gray);

text-align: right;

}

**CHAPTER 7**

**SYSTEM TESTING**

**7.1 TESTING TECHNIQUES**

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding an yet undiscovered error. A successful test is one that uncovers an yet undiscovered error. Any engineering product can be tested in one of the two ways:

**7.2 WHITE BOX TESTING**

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

* Flow graph notation
* Kilometric complexity
* Deriving test cases
* Graph matrices Control

**7.3 BLACK BOX TESTING**

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

**7.4 SOFTWARE TESTING STRATEGIES:**

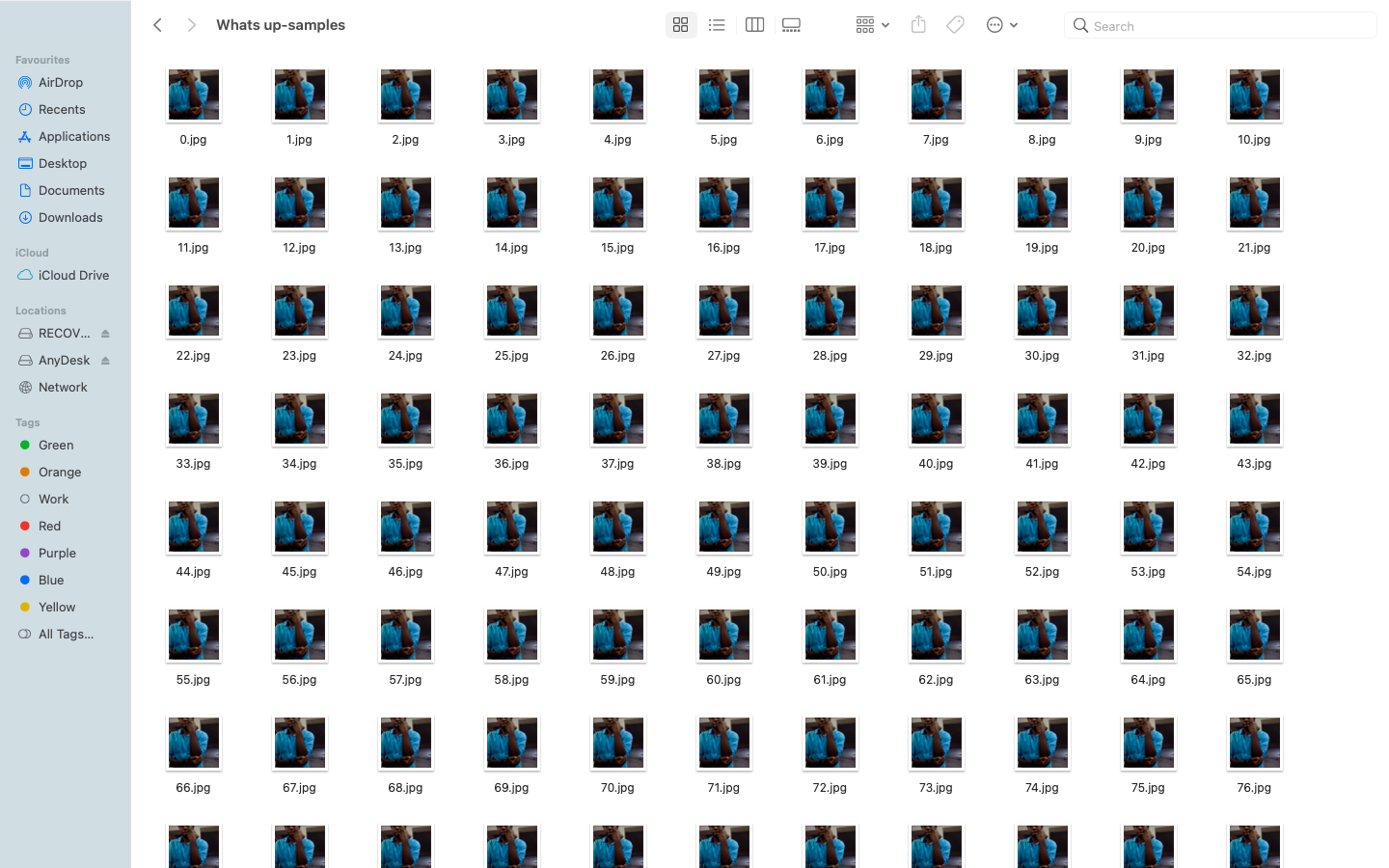
A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

Testing begins at the module level and works “outward” toward the integration of the entire computer based System

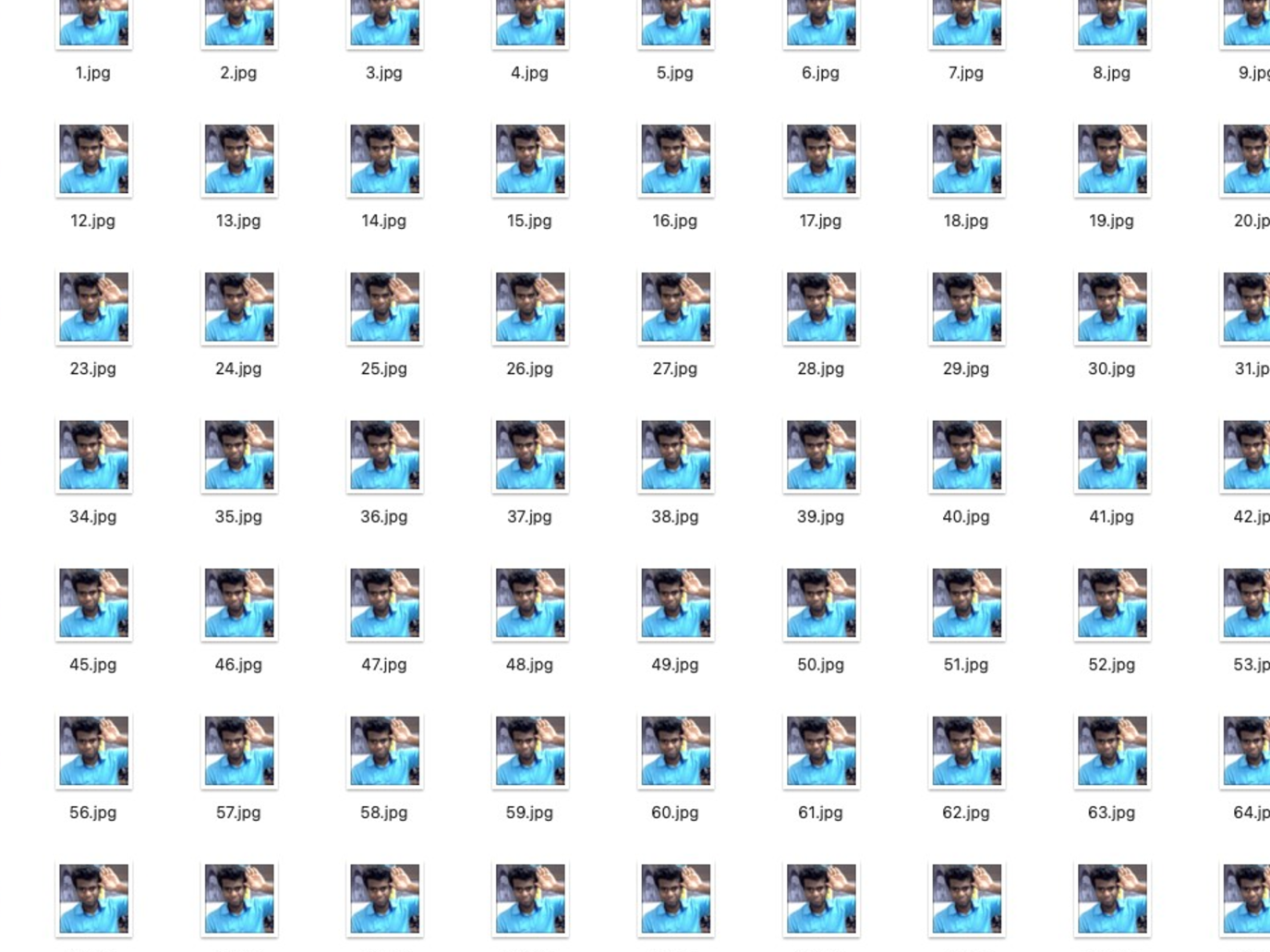
|  |  |  |  |
| --- | --- | --- | --- |
| **Test**  **Case No.** | **Action** | **Expected Output** | **Actual Output** |
| 1 | Number of characters of Roll No. is equal to  12 characters | Login button is enabled | Login button is enabled |
| 2 | Number of characters of Roll No. is less  than 12 characters | Login button is disabled | Login button is disabled |
| 3 | Number of characters of Roll No. is greater  than 12 characters | Subsequent characters are not accepted | Subsequent characters are not accepted |
| 4 | Click Help | Displays Help screen | Displays Help screen |

**CHAPTER 8**

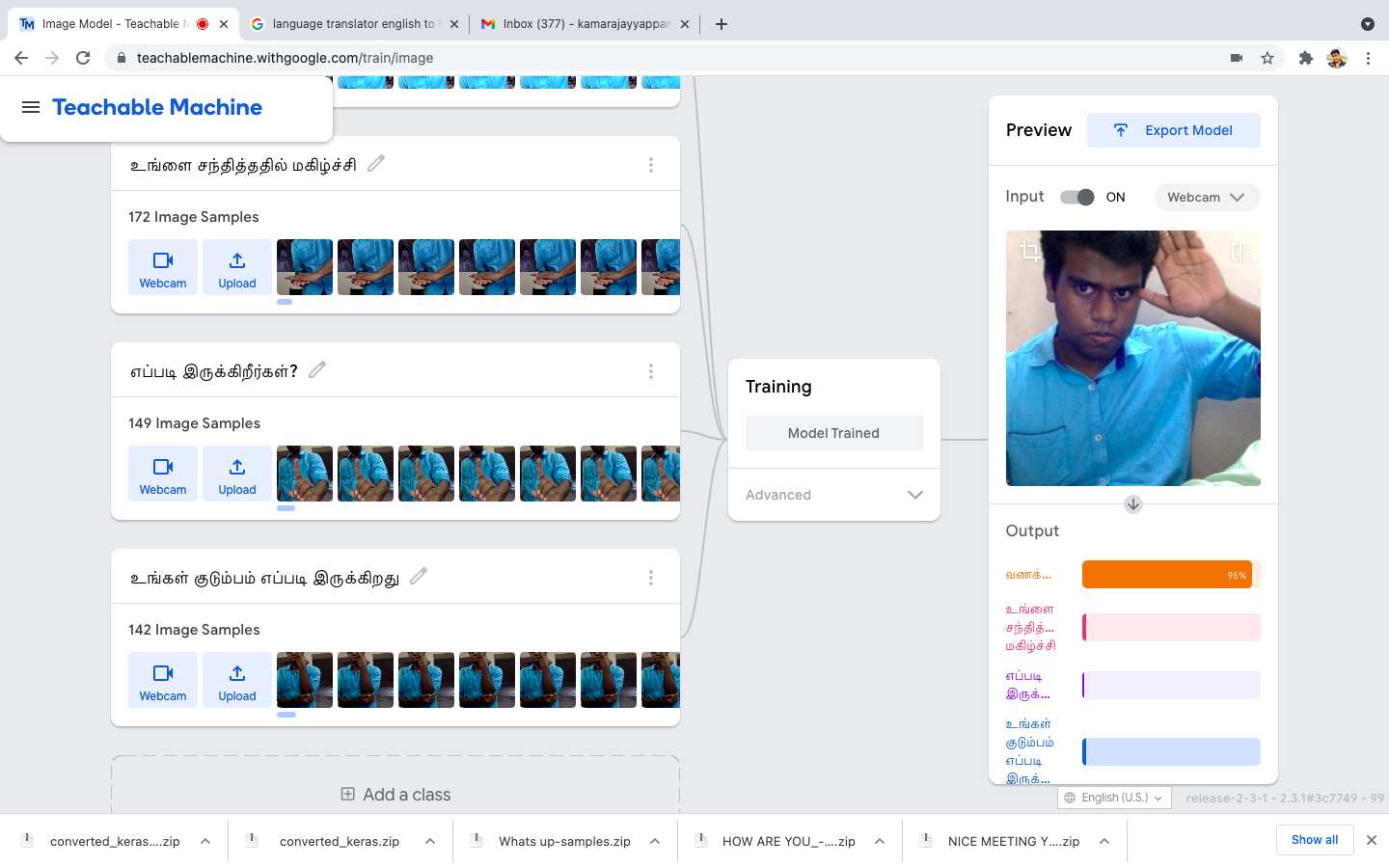
**SCREENSHOTS AND OUTPUTS**



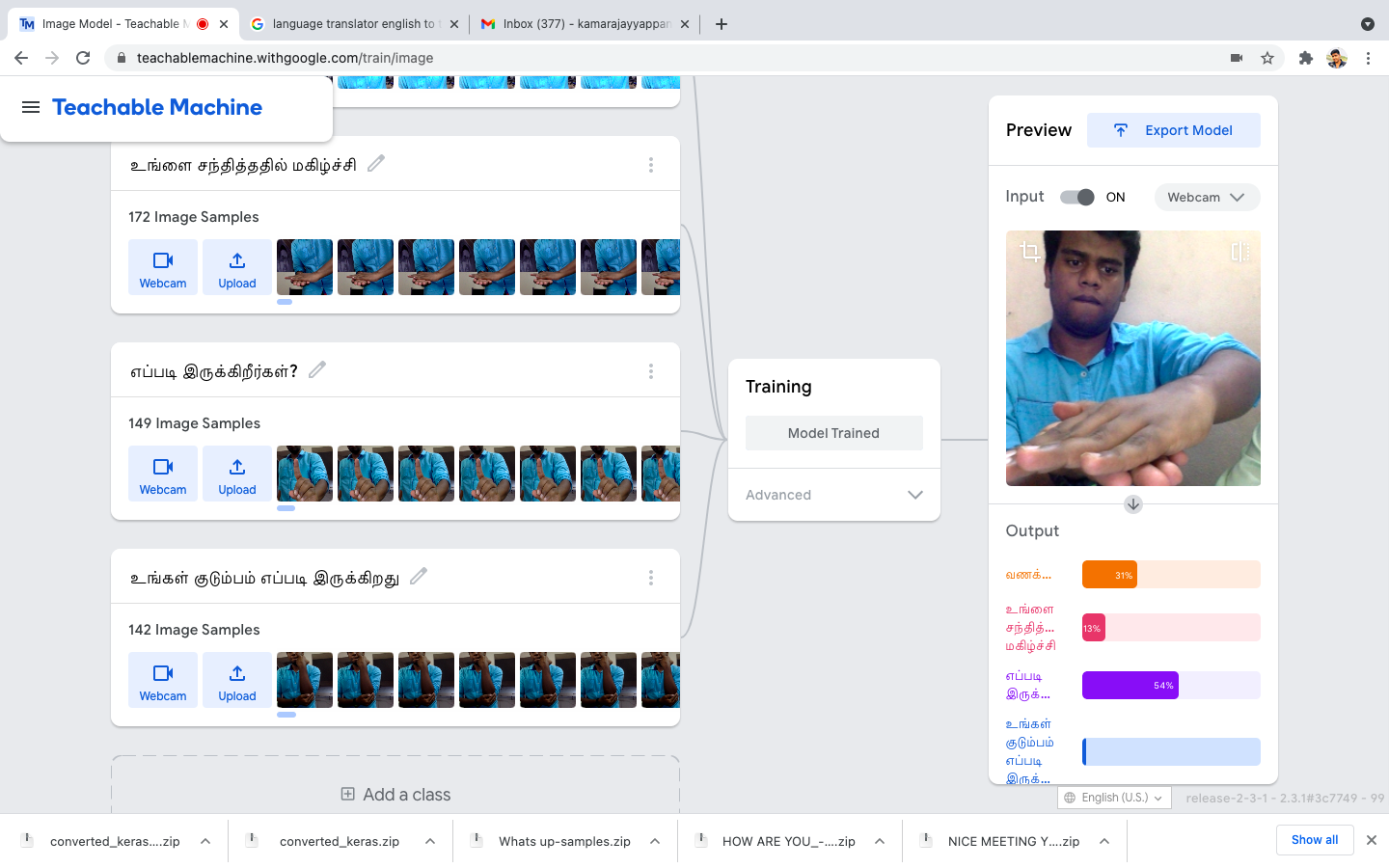
**Fig 9: Dataset Images-1**



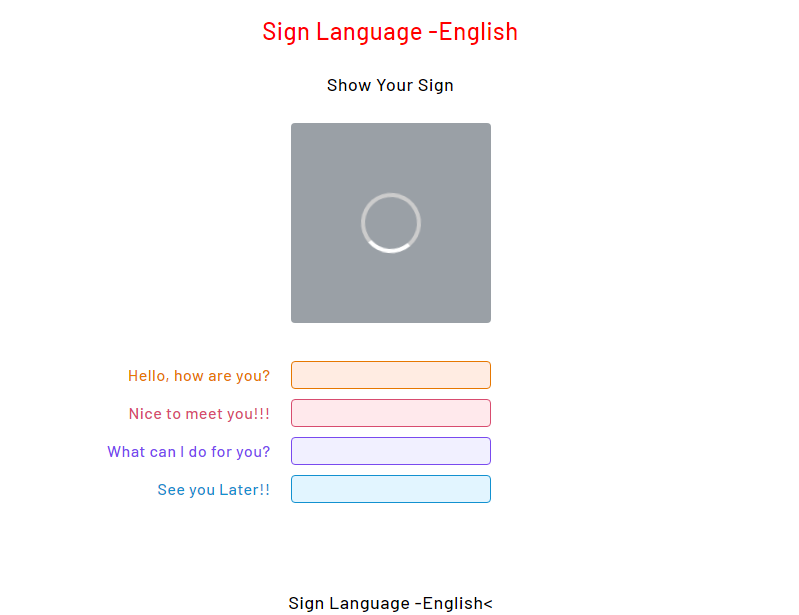
**Fig 10: Dataset Images-2**



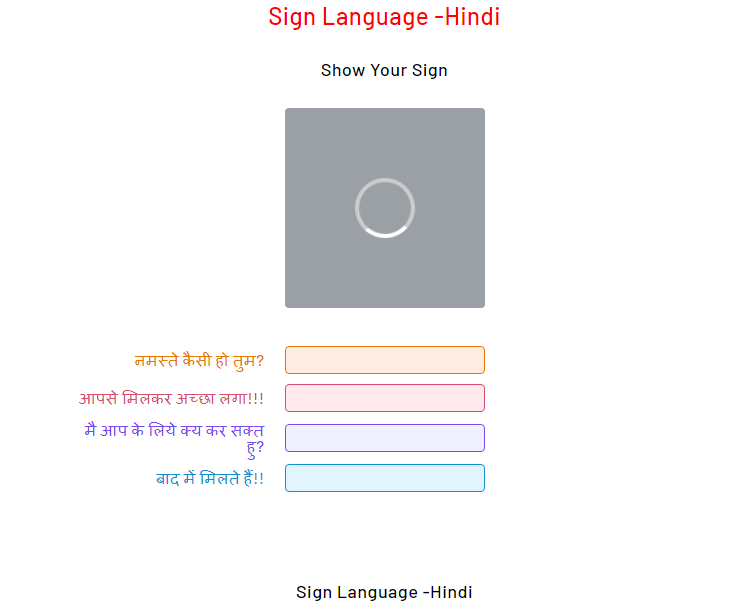
**Fig 11: Training Model**



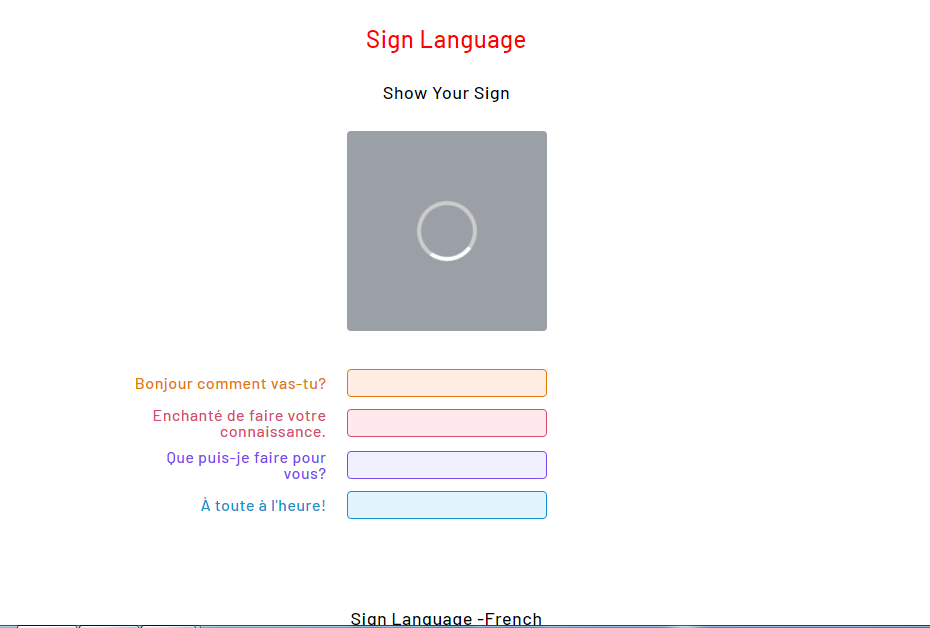
**Fig 12: Training Gestures**

****

**Fig 13: English WebUI**

****

**Fig 14: Hindi WebUI**

****

**Fig 15: French WebUI**

**CHAPTER 9**

**CONCLUSION AND FUTURE ENCHANCEMENTS**

The results also showed that the gesture recognition application was quite robust for static images. However, the video version was enormously affected by the amount of illumination, such that was necessary to check and adjust the HSV values for skin color when starting the program to get the proper output. Sometimes the adjustment was difficult to do because of the lighting conditions and the amount of objects in the background. The application was very susceptible to noise on the video stream. Slight hand movements could affect gesture recognition. Nevertheless, if the hand is steady enough for long enough; the program outputs the correct command. It was also observed that while the program was executing there were memory leaks. Attempts to remedy the problem were made by using the Open CV functions to release memory. Despite this, the leaks continued. Perhaps the leaks were due to the implementation of Open CV functions for the sequences behind the scenes.

**CHAPTER 10**

**REFERENCES**

* Meenakshi Panwar, Pawan Singh Mehra, Hand Gesture Recognition for Human Computer Interaction, International Conference on Image Information Processing India (2011).
* RafiqulZaman Khan and Noor Adnan Ibraheem, Comparitive Study of Hand Gesture Recognition System, AIRCC Digital Library – 2012.
* Arpita Ray Sarkar, G. Sanyal, S. Majumder, Hand Gesture Recognition Systems: A Survey, International Journal of Computer Applications Volume 71– No.15, May 2013.
* Manjunath A E, Vijaya Kumar B P, Rajesh H, Comparative Study of Hand Gesture Recognition Algorithms, International Journal of Research in Computer and Communication Technology, Vol 3, Issue 4, April- 2014.
* Dnyanada R Jadhav, L. M. R. J Lobo, Navigation of PowerPoint Using Hand Gestures, International Journal of Science and Research (IJSR) 2015.
* Ruchi Manish Gurav, Premanand K. Kadbe, Real time finger tracking and contour detection for gesture recognition using OpenCV, IEEE Conference May 2015, Pune India.
* Pei Xu, Department of Electrical and Computer Engineering, University of Minnesota, A Real-time Hand Gesture Recognition and Human-Computer Interaction System, Research Paper April 2017.
* P.Suganya, R.Sathya, K.Vijayalakshmi , Detection and Recognition of Gestures To Control The System Applications by Neural Networks, International Journal of Pure and Applied Mathematics 2018.
* Md. Hasanuzzaman, T. Zhang, V. Ampornaramveth, H. Gotoda, Y. Shirai, and H. Ueno,“Adaptive visual gesture recognition for human-robot interaction using a knowledge-basedsoftware platform”, Robotics and Autonomous Systems, vol. 55, Issue8,31August2007,pp.643-657. http://dx.doi.org.ezproxy.massey.ac.nz/10.1016/j.robot.2007.03.002
* R.C. Gonzales and R.E. Woods, Digital Image Processing, Prentice Hall, 2008.
* M. Sonka, V. Hlavac, and R. Boyle, Image Processing, Analysis, and Machine Vision, PWS Publishing, 1999.
* R. Szeliski, Computer Vision: Algorithms and Applications, Springer, 2011.
* S.E. Umbaugh, Digital Image Processing and Analysis, CRC Press, 2011.
* S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 2010.
* Open CV, http://www.willowgarage.com/pages/software/opencv, accessed 11 May 2011.
* A.L.C. Barczak, Computer Vision: Study Guide, IIMS, Massey University, 2010.
* G. Bradsky and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, O'Reilly, 2008.
* J.R. Parker, Algorithms for Image Processing and Computer Vision, Wiley Publishing, 2011
* R. Mukundan and K.R.Ramakrishnan, Moment Functions in Image Analysis, World Scientific, 1998.
* M.K. Hu, “Visual pattern recognition by moment invariants,” IRE Transactions on Information Theory, vol. 8, pp. 179-187, 1962.
* Y. Freund and R.E. Schapire, “A short introduction to boosting,” Journal of Japanese Society.