REAL TIME ASSISTIVE SYSTEM FOR DEAF AND DUMB COMMUNITY

The Project submitted to

Sant Gadgebaba Amravati University, Amravati

Towards partial fulfilment of the Degree of

Bachelor of Engineering

In

Information Technology

Guided by
Prof. Ms. P.V. Kale

Submitted by
Mr. Ashish Dandade
Mr. Pranav Tayade
Mr. Rushikesh Patil
Mr. Jayant Mitkari

DEPARTMENT OF INFORMATION TECHNOLOGY
SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGINEERING, SHEGAON (M.S.)

2022- 2023

04

SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGINEERING, SHEGAON

2022-2023

CERTIFICATE

This is to certify that **Mr. Ashish Dandade, Mr. Pranav Tayade, Mr. Rushikesh Patil, Mr. Jayant Mitkari** students of final year B.E. (Information Technology) in the year 2022-2023 of the Information Technology Department of this institute have completed the project work entitled "**Real Time Assistive System for Deaf and Dumb Community**" based on syllabus and has submitted a satisfactory account of his/her work in this report which is recommended for the partial fulfilment of the degree of Bachelor of Engineering in Information Technology.

Prof. Ms. P.V.Kale
(Project Guide)

Dr. A. S. Manekar
Head of the Department
SSGMCE, Shegaon

Dr. S. B. Somani
Principal
SSGMCE, Shegaon

SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGINEERING, SHEGAON

2022-2023

CERTIFICATE

This is to certify that the project work entitled "**Real Time Assistive System For Deaf And Dumb Community**" submitted by **Mr. Ashish Dandade, Mr. Pranav Tayade, Mr. Rushikesh Patil, Mr. Jayant Mitkari** students of final year B.E. (Information Technology) in the year 2022-2023 of the Information Technology Department of this institute, is a satisfactory account of there work based on the syllabus which is approved for the award of the degree of Bachelor of Engineering in Information Technology.

Internal Examiner External Examiner

Date: Date:

ACKNOWLEDGEMENT

The real spirit of achieving goals through the way of excellence and lustrous discipline. We would have never succeeded in completing our task without the cooperation, encouragement and help provided to us by various personalities.

We would like to take this opportunity to express our heartfelt thanks to our guide ***Prof. Ms P.V. Kale*** *for her esteemed guidance and encouragement, especially through difficult times. Her suggestions broaden our vision and guide us to succeed in this work. We are also very grateful for her guidance and comments while studying part of our project and learning many things under her leadership.*

We would also like to extend our sincere thanks to ***Prof. F.I. Khandwani****, Project-In-Charge for his valuable support and feedback during the entire course of the project.*

We also extend our thanks to ***Dr. A. S. Manekar****, Head of Information Technology Department, Shri Sant Gajanan Maharaj College of Engineering, Shegaon for providing us with a variety of opportunities and inspirations to gather professional knowledge and material that made us consistent performers.*

We also extend our thanks to ***Dr. S. B. Somani,*** *Principal, Shri Sant Gajanan Maharaj College of Engineering, Shegaon for providing us the infrastructure and facilities without which it was impossible to complete this work.*

Also, we would like to thank all teaching and non-teaching staff of the department for their encouragement, cooperation and help. Our greatest thanks to all those who wished us success, especially parents and friends.

Student Names

Mr. Ashish Dandade
Mr. Pranav Tayade
Mr. Rushikesh Patil
Mr. Jayant Mitkari

`

ABSTRACT

People communicate with other people by using words that come from various tongues, but if somebody is deaf or dumb and incapable of listening or talk? People who are deaf or dumb must use a particular sign for communication purposes. A person with translation skills is required since not everybody grasps sign language, which breaks down interaction between sensing of deaf/dumb people. The deaf/dumb people need to communicate using a certain sign language. As sign language is not understandable by everyone, a translator is needed to make up communication between normal and deaf/dumb people. Practically, translators are not available all the time hence a beneficial impact on the social lives of these hearing/speech disabled people can be brought by a system that recognizes hand gestures and performs the task same as translators.

An automated model that identifies hand gestures and conduct the same job as translators could have a beneficial impact on the social lives of these hearing and speech impaired people because interpreters aren't always readily available. Applying the YOLOv5 machine learning model, hand motions have been learned and predicted. It will have many applications such as, online communications, Touchless operating, Human Computer Interaction (HCI).

Hand gesture recognition research classified as Glove based and vision based; Glove based system affects the natural signing ability. Vision based category is not using any devices or gloves, is the most natural way. Data acquisition can be done with a camera. Conventional methods of gesture recognition is time consuming, so the usage of YOLOv5 is to prevent the segmentation or detection stage which favors faster real-time process.

Here the development of the model is in such a way it can help us to recognition and implement it. For such developments these models will be working with YOLOv5 and Back Propagation methodologies which can help for the easy working of these models.

Keywords: *Machine Learning, YOLOv5, Sign Language, Python*.

`

TABLE OF CONTENTS

`

`

LIST OF FIGURES

LIST OF TABLE

1. INTRODUCTION

1.1 Preface

In World, there are 63 million deaf and dumb people. Among them 2.4 million of people live in India. These individuals lack the conveniences that a typical person would have. The main cause of this is a lack of communication since people cannot talk and deaf people cannot listen. Sign language is used by those who have difficulty in hearing or speaking as a means of communication. In sign language, people express their thoughts and feelings via nonverbal gestures. As shown in Figure 1.1 However, non-signers find it very difficult to understand, thus skilled sign language interpreters are needed during training sessions as well as sessions for legal, medical, and educational uses.

Over the preceding five years, there has been an increase in demand for translation services. More techniques are now readily available, including high-speed video remote human interpretation. As a result, they will provide a service that is easy to use but has significant disadvantages for sign language interpretation. To solve this, we use a combination of two models to differentiate between sign language motions. Using video footage from the distinct Sign Language Dataset, we train the computer to detect motions. A range of motions that were performed repeatedly in various video and context settings are included in the collection. For convenience, the videos have been captured at a consistent frame rate. In contrast to the more traditional "sliding window" technique, which frequently requires numerous classifications runs for picture segments, our "You Only Look Once" approach analyses input images whole. This method speeds up processing since fewer separate assessments are required, and accuracy is increased because the whole neural network has access to the global visual context.

A hearing disorder is a condition when any part of the ear or sensory system isn't working in the right way. The hearing disorder may be broadly classified into three types. Sensorineural hearing disorder: This disorder causes a prominent form of hearing impairment. It is caused when the receptor nerves and hair cells are damaged mainly because of age, noise. Sensorineural hearing disorder affects the pathways from receptor to brain. Majority of the times, this disorder can't be rectified medically or surgically, but can be assisted by the use of hearing aids [9].

Conductive hearing disorder: This kind of hearing loss is caused as a result of blockages within the outer or bodily cavity due to fluids, tumors, earwax. This blockage halts the sound waves from attending the receptor. This disorder can often be healed surgically or cured with medication [9].

Auditory Neuropathy Spectrum Disorder (ANSD): This disorder is caused when the sound waves vibrating the cochlea are not organized or the hearing nerve doesn't process sound even if receptor encounters sound normally. There are possibly many reasons for this disorder like directly inherited through genes from the ancestors or due to trauma [10]. ANSD can be induced in several cases such as lack of oxygen at birth, blood transfusion is required, premature birth, at the time of birth, ototoxic drug exposure, immune disorders, mumps caused by infection and many more [10].

Sign language is communication that involves body language and lip movements without using sound. Usually, the sign language used to mean communication for hearing impaired and deaf people who have limited oral communication. Therefore, like language in general, sign language also uses standard provisions that are easy for everyone to understand. Usually, hearing impaired and deaf people are accustomed to memorizing and understanding hand movements as sign language to communicate with their fellow people as this is the most effective communication needs. However, the problem arises if the deaf person wants to communicate with the ordinary person. They cannot understand the sign language. For people who can communicate orally, the use of sign language is not a concern, so it is not easy to learn it. Sign language is sawed as a sequential image that has the characteristics of specific symbols. The rapid development of machine learning, such as Deep Learning allows the system to learn and remember patterns of hand movements and facial expressions as certain symbols in sign language.

Formulating algorithms and systems to discover a series of formed signs and to recognize their meaning in the form of content or verbal communication is referred to as sign language recognition. Sign language recognition (SLR) follows a line of investigation involving artificial neural networks, computer vision and linguistics and etc. Since the signs are done through hands include both single hand and two hands, facial expressions include eyes and eyebrows and other parts of the body and it is a wide-ranging issue.

A sign language recognition (SLR) system is categorized into two groups [5]: vision-based systems and glove based systems. The later faction necessitates end user to put on gloves. The glove facilitates the recognition to evade or make simpler the segmentation and detection. On the other hand, users have to bear a hardware gadget that formulates them. Besides, the glove-based System possibly mislay the emotional expression of the signer from facial expression, essential for sign language recognition. On the other hand, the earlier faction, rely on vision-based techniques where the sequence of signs are captured with video camera, is easier for users. Nevertheless, it is difficult to precisely segment and track.

Recently, visual communications like augmented reality and virtual reality has attracted attention since many companies have introduced futuristic devices for the company's growth. For example, smart glasses for realistic features and many applications that have been developed in many companies. Virtual glass and augmented devices displays have also been introduced and are being used in futuristic applications. Hand gesture recognition is widely used as an interface to give commands to these reality-controlled devices. The system that acquires a perfect balance between the concepts considered as well performed and super powerful systems. Gestures can act as communication barrier between the human and machines. Also, can develop between people who cannot communicate properly.

Figure. 1.1 Signs Deaf and Dumb people use while communicating.

There are various type of signs deaf and dumb people using while communication some are above mentioned like YES, NO, THANK YOU, SORRY shown in above figure.

1.2 Statement of problem

A World Health Organization report says around 63 million people in India suffer from either complete or partial deafness. The country only has about 700 schools which teach sign language. A collection of visual cues, hand gestures and devices has found to be communicative mechanism by people with hearing or speech impairments to interact with each other. However, it is difficult for normal people to understand their signs and message.

1.3 Objectives of Project

The objective of this project can be summarized as follows:

[1] The proposed system is used to recognize the real-time signs. Hence it is useful for hearing and speech impaired people to communicate with normal people.

[2] Designing and Developing a Real Time assistive system which will have a camera unit for capturing the gestures of the hearing and speech impaired people and will be designed as a portable unit for more convenience of the users.

[3] To reduce the level of human intervention necessary in extracting data from document.

[4] System will give output in text format at real time.

[5] It can be embedded in web app.

1.4 Scope and Limitations of the Project

The Project Real Time Assistive System for Deaf and Dumb Community can recognize the sign done by deaf and dumb community and convert it into text format.[2] The Project also has the potential to convert sign language in audio format.

The Project Scope includes:

Collecting various sign languages of deaf and dumb community to build a comprehensive database for model training.

Developing and testing machine learning algorithms and models that can accurately predict sign languages done by deaf and dumb people.

Incorporating real-time data streams into the models to enhance the accuracy and timeliness of predictions and recommendations.

The future scope for a sign language detection system can be vast, as there are many possible areas for improvement and expansion. Here are some potential future directions for the development of a sign language detection system:

a. Multi-modal input: A sign language detection system could be expanded to include other modalities such as facial expressions or body language. This could enable a more nuanced understanding of sign language communication and improve accuracy.

b. Machine learning: As machine learning algorithms continue to improve, there is potential for a sign language detection system to become more accurate and reliable. Additionally, machine learning could be used to improve the system's ability to recognize signs from different sign languages.

c. Augmented reality: The use of augmented reality (AR) could enable the system to overlay text or images on the real world to provide additional context for the communication. This could be particularly useful for educational or training purposes.

d. Gesture recognition for non-sign languages: The technology used in a sign language detection system could potentially be applied to recognizing other gestures or movements, such as those used in dance or sports.

e. Collaborative communication: A sign language detection system could be expanded to enable communication between multiple users, with the system providing translation between different sign languages or spoken languages.

f. Mobile applications: A mobile application could be developed to enable users to access the sign language detection system on the go, allowing for greater accessibility and convenience.

g. Integration with other assistive technologies: A sign language detection system could be integrated with other assistive technologies such as hearing aids or cochlear implants to provide a more comprehensive communication solution for people with hearing impairments.

Limitations:

- **Poor Image Quality Limits Hand Recognition's Effectiveness.**
  Image quality affects how well Hand-recognition algorithms work. The image quality of scanning video is quite low compared with that of a digital camera. Even high-definition video is, at best, 1080p (progressive scan); usually, it is 720p. These values are equivalent to about 2MP and 0.9MP, respectively, while an inexpensive digital camera attains 15MP. The difference is quite noticeable.

- **Small Image Sizes Make Signs Recognition More Difficult**
  When a hand -detection algorithm finds a face in an image or in a still from a video capture, the relative size of that hand compared with the enrolled image size affects how well the face will be recognized. An already small image size, coupled with a target distant from the camera, means that the detected face is only 100 to 200 pixels on a side. Further, having to scan an image for varying face sizes is a processor intensive activity. Most algorithms allow specification of a hand signs size range to help eliminate false positives on detection and speed up image processing.

- **Different hand Angles Can Throw Off Hand Recognition's Reliability.**
  The relative angle of the target's hand influences the recognition score profoundly. When a face is enrolled in the recognition software, usually multiple angles are used Anything less than a frontal view affects the algorithm's capability to generate a template for the hand. The more direct the image (both enrolled and probe image) and the higher its resolution, the higher the score of any resulting matches.

- **Data Processing and Storage Can Limit Hand Recognition Technology.**
  Even though high-definition video is quite low in resolution when compared with digital camera images, it still occupies significant amounts of disk space. Processing every frame of video is an enormous undertaking, so usually only a fraction (10 percent to 25 percent) is actually run through a recognition system. To minimize total processing time, agencies can use clusters of computers. However, adding computers involves considerable data transfer over a network,

which can be bound by input and output restrictions, further limiting processing speed. On a too short sequence, or when the target remains static cannot learn the person specific mouth behavior. In this case, temporal aliasing can be observed, as the target space of the retrieved mouth samples is too sparse. Another limitation is caused by our hardware setup (webcam, USB, and PCI), which introduces a small delay of ≈ 3 frames. Specialized hardware could resolve this, but our aim is a setup with commodity hardware. The requirement of a short scanning sequence based on an RGB-D camera. Failure cases that stem from extreme head rotations and occlusions in the input stream of the source actor.

1.5 Organization of the Project

The project is organized as follows:

1. Chapter 1 gives Introduction about the project.
2. Chapter 2 gives Literature survey of the project.
3. Chapter 3 provides analysis of project.
4. Chapter 4 provides design phase of project.
5. Chapter 5 provides how project is implemented.
6. Chapter 6 gives conclusion with future scope of the project.

2. LITERATURE SURVEY

Paper 01:

"Design of Sign Language Recognition Using E-CNN" By Citra Suardi and Anik Nur Handayani 2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT) | 978-1-6654-0514-0/20/$31.00 ©2021 IEEE | DOI: 10.1109/EIConCIT50028.2021.9431877. [1]

Description: Most people do not understand sign language, so they need a bridge for the community to be able to communicate with deaf people. Technology that continues to develop and continues to strive to help humans, can be a solution that can be used to create a communication bridge between the community and deaf people, the use of technology that can be used is the use of image processing technology as a translator tool. Image processing can translate images into text. In the implementation of digital image processing, it will use the hand key point library, where the hand key point library is a library that will detect the location of the hand in each image, but as it is known, image processing cannot stand alone as a data processor but requires an algorithm that functions as a classification tool. The Convolutional Neural Network (CNN) algorithm in the Deep Learning method can be a classification tool, with the ability of the Convolutional Neural Network (CNN) to learn several things. And according to several previous studies that combining several algorithms can increase the accuracy value. In this study, a trial of combining CNN models using the Ensemble method has been successfully carried out with the results being able to increase the accuracy value to 99.4%. So that the results of the research can be summarized that using Ensemble can increase the higher accuracy value.

Paper 02:

"A New Benchmark on American Sign Language Recognition using Convolutional Neural Network" By Md. Moklesur Rahman∗ , Md. Shafiqul Islam 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), 24-25 December, Dhaka 978-1-7281-6099-3/19/$31.00 © 2019 IEEE. [10]

Description: According to the World Health Organization (WHO) , the number of people having hearing or listening disability increased from 278 million in 2005 to 466 million in early 2018. It is assumed that this number will be increased to 400 million by 2050t this number will be increased to 400 million by 2050. This deaf community uses a set of signs to express their language (called sign language), which is different for different nations. In other words, a sign language (SL) is a nonverbal communication language, which utilizes visual sign patterns made with the hands or any parts of the body, used primarily by the people who have the disability of hearing and/or listening. Sign languages (SLs) are fullfledged natural languages with their own lexicon and grammar.

A convolutional neural network (CNN), one of the most popular deep learning algorithms, comprising of convolutional layers and then following by one or more fully connected layers, was by Vallian et al.. From a computer science perspective, a CNN is a set of digital filters whose weights are estimated during the learning phase. Naturally, there are more complex processes occurring in human brain. Following this analogy, each convolutional layer extracts features from training data. A CNN convolves learned features with input data, utilizes convolutional layers, and turning this architecture into well suited form to process data. CNNs learn to detect different features of data using multiple hidden layers. Every hidden layer increases the difficulty of the learned data features. The problems with the existing methods for ASL recognition are that they have reported their study on a specific dataset (most of the cases), rare comparison of the methods on a common dataset. So, cannot compare in which degree a method is better/worse than another or which dataset does possess proper variation in samples for sufficient training of a classification model. To address these problems, int this paper considered four publicly available ASL datasets on which several good works have been reported. This paper studied the performance of the model on each dataset, when trained using the training set and tested using the test set (if separate train and test sets are available or using the 10-fold cross-validation). The performance of the method has also been justified using cross dataset i.e., trained using one dataset and tested on a different dataset. In addition, the performance has been compared with previous methods on the same dataset.

Paper 03:

"Classification of American Sign Language by Applying a Transfer Learned Deep Convolutional Neural Network" By Md. Mehedi Hasan, Azmain Yakin Srizon 2020

23rd International Conference on Computer and Information Technology (ICCIT), 19-21 December, 2020 978-1-6654-2244-4/20/$31.00 ©2020 IEEE. [5]

Description: To defeat the connection passage among deaf-mute people, sign literature is employed that is the common exercised literature among the deaf-mute community to interact among people and bestow opinions. Deaf-mute is an expression that is exercised historically to identify a person who is either deaf or both deaf and cannot speak at the same time, and in both cases, sign language is the process of interaction for them. Sign language is a kind of literature that employs visual-manual procedures to interact or carry meaning. Like natural languages, sign languages possess their grammar and vocabulary. Despite holding notable associations between sign languages, they are not extensively the same and mutually acknowledged. Being stated that, many researchers have contributed significantly for the accurate recognition of sign language characters. However, in this research, paper specifically focused on American sign language character recognition. Having 250,000 to 500,000 persons in the deaf community of Americans and some Canadians who utilize American sign language, this was an obvious opportunity for research. Firstly, the data was split into the train set and test set. 80% of data was kept in the train set and the rest of the 20% data was kept testing set. Then, the modified inceptionV3 architecture was employed to train set. Training accuracy and validation accuracy of architecture are illustrated. On the other hand, the training and validation loss of our architecture. illustrates the confusion matrix of ASL detection. Table-1 illustrates the class-wise accuracy, precision, recall, f-score, and support of each of the classes under consideration. Our architecture produced overall correctness of 98.81% for American Sign Language dataset. illustrates the comparison among our work and notable previous works. From it can be noticed that our architecture outperformed all the previous approaches by a notable margin, hence, our model can recognize the considered classes more accurately.

Paper 04:

"Convolutional Neural Network Hand Gesture Recognition for American Sign Language" Shruti Chavan, Xinrui Yu and Jafar Saniie 2021 IEEE International Conference on Electro Information Technology (EIT) | 978-1-6654-1846-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/EIT51626.2021.9491897.[2]

Description: With the advancements in the computer vision technology, learning and using sign languages to communicate with deaf and mute people has become easier. Exciting research is ongoing for providing a global platform for communication in different sign languages. In this paper, author present a Deep Learning based approach to recognize a sign performed in American Sign Language by capturing an image as input. Convolutional neural networks are specifically designed to make inference from visual data such as images and videos. The features are extracted and learnt to tarin the model, which gives better recognition accuracy compared to conventional Machine Learning algorithms. CNNs have numerous applications in the field of Signal Processing, robotics, medical imaging, data analysis, Business Intelligence, etc. The learning from unaltered and smaller dataset with CNNs yields surprisingly better results. The system can predict the signs of 0 to 9 digits performed by the user. By utilizing image processing to convert RGB data to grayscale images, efficient reduction is achieved in the storage requirements and training time of the Convolutional Neural Network. The objective of the experiment is to find a mix of Image Processing and Deep Learning Architecture with lesser complexity to deploy the system in mobile applications or embedded single board computers. The database is trained from scratch using smaller networks as LeNet-5 and AlexNet as well as deeper network such as Vgg16 and MobileNet v2. The comparison of the recognition accuracies is discussed in the paper. The final selected architecture has only 10 layers including a dropout layer which boosted the training accuracy to 91.37% and testing accuracy to 87.5%.

Paper 05:

"American Sign Language Recognition using Deep Learning and Computer Vision" By Kistij Bantupalli and Ying Xie 2018 IEEE International Conference on Big Data (Big Data) 978-1-5386-5035-6/18/$31.00 ©2018 IEEE. [12]

Description: Speech impairment is a disability which affects an individual's ability to communicate using speech and hearing. People who are affected by this use other media of communication such as sign language. Although sign language is ubiquitous in recent times, there remains a challenge for non-sign language speakers to communicate with sign language speakers or signers. With recent advances in deep learning and computer vision there has been promising progress in the fields of motion and gesture recognition using deep learning and computer vision-based techniques. The focus of this work is to create a vision-based application which offers sign language translation to text thus aiding communication between signers and non-signers. The proposed model takes video sequences and extracts temporal and spatial features from them. then use Inception, a CNN (Convolutional Neural Network) for recognizing spatial features. In paper use a RNN (Recurrent Neural Network) to train on temporal features. The dataset used is the American Sign Language Dataset. Index Terms—computer science, machine learning, computer vision, sign language Sign language is a form of communication used by people with impaired hearing and speech. People use sign language gestures as a means of non-verbal communication to express their thoughts and emotions. But non-signers find it extremely difficult to understand, hence trained sign language interpreters are needed during medical and legal appointments, educational and training sessions. Over the past five years, there has been an increasing demand for interpreting services. Other means, such as video remote human interpreting using high-speed Internet connections, have been introduced. They will thus provide an easy-to-use sign language interpreting service, which can be used, but has major limitations.

To address this, in the paper an ensemble of two models to recognize gestures in sign language. use the custom American Sign Language Dataset for video data for training the model to recognize gestures. The dataset has different gestures performed multiple times giving us variation in context and video conditions. For simplicity, the videos are recording at a common frame rate. paper propose to use a CNN (Convolutional Neural Network) model named Inception to extract spatial features from the video stream for Sign Language Recognition (SLR). Then, by using a LSTM (Long Short-Term Memory) a RNN (Recurrent Neural Network) model.

Paper 06:

"Hand Gesture Detection based Real-time American Sign Language Letters Recognition using Support Vector Machine". By Xinyun Jiang and Wasim Ahmad. 978-1-7281-3024-8/19/$31.00 ©2019 IEEE DOI 10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00078. [7]

Description: Sign language is an indispensable communication means for deaf-mute people because of their hearing impairment. At present, sign language is not popular communications method among hearing people, so that the majority of the hearing are not willing to have a talk with the deaf-mute, or they have to spend much time and energy trying to figure out what the correct meaning is. Sign Language Recognition (SLR), which aims to translate sign language to people who know few about it in the form of text or speech, can be said to be a great help to deaf-mute and hearing people to communicate. In this study, a real-time vision-based static hand gesture recognition system for sign language was developed. All data is collected from a USB camera connected to a computer, and no auxiliary items (such as gloves) were required. The App is based on skin color algorithm in HSV color space to find the Region of Interest (ROI), where hand gesture is. After completing all pre-processing work, 8 features were extracted from each sample using Principal Component Analysis (PCA). The recognition machine learning approach used was based on Support Vector Machine (SVM). The experimental results show that this system can distinguish B, D, F, L and U, these five American sign language hand gestures, with the successful rate of about 99.4%.

Paper 07:

"Hand Gesture Recognition Based on Deep Learning" By Jing-Hao Sun, Ting-Ting Ji and Shu-Bin Zhang 978-1-5386-7302-7/18/$31.00 ©2018 IEEE. [13]

Description: rapid development of computer vision, the demand for interaction between human and machine is becoming more and more extensive. Since hand gestures are able to express enriched information, the hand gesture recognition is widely used in robot control, intelligent furniture and other aspects. The paper realizes the segmentation of hand gestures by establishing the skin color model and AdaBoost classifier based on hear according to the particularity of skin color for hand gestures, as well as the denaturation of hand gestures with one frame of video being cut for analysis. In this regard, the human hand is segmented from the complicated background, the real-time hand gesture tracking is also realized by Cam Shift algorithm. Then, the area of hand gestures which has been detected in real time is recognized by convolutional neural network so as to realize the recognition of 10 common digits. Experiments show 98.3% accuracy.

Paper 08:

"Sign Language Alphabet Reorganization Using Convolutional Neural Network" Proceedings of the Fifth International Conference on Intelligent Computing and Control Systems (ICICCS 2021) IEEE Xplore Part Number: CFP21K74-ART; ISBN: 978-0-7381-1327-2.[3]

Description: Sign Language plays an indispensable role in the lives of people having speaking and hearing disabilities. Recognition of American Sign Language using Computer Vision is very challenging due to its increasing complexity and high intraclass variations. In this paper, convolutional neural networks (CNNs) are used to recognize the ASL Alphabets. Collecting data is one of the most essential and vital parts of any exploration. It is essential to collect data that is relevant and satisfies the requirements needed for the research. The data should contain readings that replicate all scenarios, including extreme cases. This helps in having more thorough and realistic observations. The data collected forms the foundation for any research and should be collected with utmost care. The research work has decided to design the dataset to match the MNIST classic dataset. Therefore, in the dataset, each element is labelled as training or test cases ranging from 0 to 25, which act as a matched map to all the alphabets in the English language ranging from A-Z. The dataset contains 27,455 training samples and 7172 testing data. This is approximately half the size of the MNSIT dataset, but it replicates it accordingly as every header row has a label of pixel1 to picxel784, which represents the grayscale values of 28x28 pixel images in which the values range from 0-255. The data collected in the MNSIT data is varied by having the same signs made by different subjects in different backgrounds. Moreover, the images were also cropped in various ways; however, the desired hand region was left unchanged. After exploring the dataset, the data is observed to be evenly balanced across the values. The wide variety of training sample values are evenly distributed, as can be seen from the visualization. This algorithm is useful to recognize it as a deep network, which is expected for the ASL alphabet classification task. Preprocessing steps of the MNIST dataset are done in the first phase. After the first phase, different important features of pre-processed hand gesture image are computed. In the final phase, depending on the properties computed or calculated in the initial phases, the accuracy and AUC score of the network model with which it can recognize the sign language Alphabets were detected. In the paper CNN network has achieved an AUC score of 0.9981 and an accuracy of 0.9963.

Paper 09:

"An Efficient Real-Time Emotion Detection Using Camera and Facial Landmarks" Binh T. Nguyen, Minh H. Trinh, Tan V. Phan and Hien D. Nguyen. 2017 Seventh International Conference on Information Science and Technology (ICIST) 10.1109/ICIST.2017.7926765.[9]

Description: Emotion recognition has many useful applications in daily lives. In this paper, present a potential approach to detect human emotion in real time. For any face detected in camera, extract the corresponding facial landmarks and examine different kinds of features and models for predicting human emotion. The experiments show that our App can naturally detect human emotion in real time and achieve an average accuracy about 70.65%.

To build a real-time emotion detection system utilize one camera of resolution 640x480 to capture human faces. There are four modules in the paper. For each frame, firstly detect all human faces and extract facial landmarks from each observed face. Then, normalize facial landmarks and calculate the corresponding features. Finally, the computed features are used as inputs of a trained classifier to predict emotion for each person. The method can be illustrated In what follows.

Paper 10:

"Sign Language Recognition Using Image Based Hand Gesture Recognition Techniques" 2016 Online International Conference on Green Engineering and Technologies (IC-GET) 978-1-5090-4556-3/16/$31.00 ©2016 IEEE.[8]

Description: Hand gesture is one of the methods used in sign language for non-verbal communication. It is most used by deaf & du mb people who have hearing or speech problems to communicate among themselves or with normal people. Various sign language systems have been developed by many makers around the world but they are neither flexible nor cost-effective for the end users. Hence in this paper introduced software which presents a system prototype that can automatically recognize sign language to help deaf and dumb people to communicate more effectively with each other or normal people. Pattern recognition and Gesture recognition are the developing fields of research. Being a significant part in nonverbal communication hand gestures are playing key role in our daily Life. Hand Gesture recognition system provides us an innovative, natural, user friendly way of communication with the computer which is more familiar to the human beings. By considering in mind the similarities of human hand shape with four fingers and one thumb, the software aims to present a real time system for recognition of hand gesture on basis of detection of some shape-based features Like orientation, Centre of mass centroid, fingers status, thumb in positions of raised or folded fingers of hand.

Paper 11:

"Research on Communication App for Deaf and Mute People Based on face reorganization Technology" By Yuan Tao and Shihang huao 2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT) | 978-1-7281-9948-1/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICCASIT50869.2020.9368771.[4]

Description: Deaf people cause inconvenience to their daily communication due to different degrees of hearing loss, and their hearing loss may also affect their ability to recognize facial emotions, thereby further affecting their interpersonal communication. Therefore, the use of emerging information technology to solve the hearing impairments of deaf and dumb people, language barriers of deaf and dumb people, regional language barriers, and emotional barriers of deaf and dumb people contributes to the smooth communication between deaf and dumb people and hearing people. And improving the education system for all has important practical significance. Based on this research background, the paper uses eye movement and event-related potential analysis methods to try to study the characteristics of deaf people's potential cognitive abilities in face emotion recognition and sign language recognition tasks from different experimental paradigms, extract the target area by means of image processing, and Use appropriate feature description, sign language video recognition technology, emotional speech synthesis technology, speech recognition technology and machine learning methods to make further accurate judgments, and finally perform sign language and emotion recognition. Finally, the APP software was tested for running. Through experiments on 630 gesture images, the recognition rate reached 94.22% and the speed reached 0. 29s/frame. The results show that the algorithm effectively improves the recognition rate and can meet the Realtime performance of dumb language communication.

Paper 12:

"Real-Time Sign Language Recognition Based on Video Stream" Proceedings of the 39th Chinese Control Conference July 27-29, 2020, Shenyang, China.[6]

Description: here are millions of deaf-dumb people in the world communicating by sign language, thus designing a sign language recognition system is very meaningful and valuable for normal people to understand them. In this paper investigate a real-time Chinese sign language recognition system. A Chinese sign language dataset is firstly created. Considering practical applications, RGB camera is used to collect video stream, instead of RGB-D camera. In order to improve the accuracy of recognition -CNN method combined with optical flow processing. The collected RGB video stream is processed by optimized dense optical flow, and then put into 3D-CNN to extract feature vectors. For practical considerations, a real-time sign language recognition system is designed, composed of artificial interaction interface, motion detection module, hand and head detection module, etc.

Paper 13:

"Sign Language Recognition Based On Hand And Body Skeletal Data" by Dimitrios Konstantinidis, Kosmas Dimitropoulos and Petros Daras, 978-1-5386-6125-3/18/$31.00 ©2018 European Union.[14]

Description: Sign language recognition (SLR) is a challenging, but highly important research field for several computer vision systems that attempt to facilitate the communication among the deaf and hearing-impaired people. In this work, an accurate and robust deep learning-based methodology for sign language recognition from video sequences. Our novel method relies on hand and body skeletal features extracted from RGB videos and, therefore, it acquires highly discriminative for gesture recognition skeletal data without the need for any additional equipment, such as data gloves, that may restrict signer's movements. Experimentation on a large publicly available sign language dataset reveals the superiority of our methodology with respect to other state of the art approaches relying solely on RGB features.

Previous works on sign language recognition were based either on direct measurement of skeletal data from obtrusive sensors and data gloves or inaccurate processing of video sequences. This paper presents a novel SLR system that attempts to overcome the limitations of previous methods by proposing the extraction and processing of hand and body skeletal data from video sequences. The experimentation on LSA64 dataset shows that our SLR system outperforms other vision based SLR approaches, despite difficulties in extracting accurate skeletal data due to occlusions. As a future work plan to test our novel SLR system in additional sign language datasets and study the contribution of image and optical flow features in the task of sign language recognition.

Paper 14:

"Indoor Speech Interaction System for Deaf-mute Based on ZigBee" Proceeding of the IEEE International Conference on Information and Automation Yinchuan, China, August 2013. 978-1-4799-1334-3/13/$31.00 ©2013 IEEE.[15]

Description: Speech recognition technology and ZigBee technology is widely implemented to improve the quality of life especially for the deaf-mute. This paper presents the design and implementation of a speech interaction system for the deaf-mute based on ZigBee wireless sensor work (WSN). In this system, one speech-recognition modules have been added to the ZigBee-based networks. The recognized messages are sent by these modules then be routed to the target device, and finally be carried out by micro control unit (MCU). Because of the limited number of users, speaker-dependent speech recognition has been used in our work. Also, this system has realized a function of sound alerts for deaf-mute which is based on many acoustic sensors. In order to help deaf-mute, perceive the sound when they are alone at home, these sensors are installed on the doors and phone. Some experimental results validate the basic functions of the App. It turns out to be convenient and effective to communicate with deaf-mute for families at home.

With the development of speech recognition technology and wireless sensor network technology, there are more and more solutions for the deaf-mute interaction. This paper presents the intelligent interaction system for deaf-mute based on ZigBee technology and speech recognition technology to be used at home. Through the theoretical analysis and experimental environment test, it has been confirmed that the design of indoor speech interaction system for deaf-mute is reasonable, the hardware system performance normal, and the software design is user-friendly that the deaf-mute can communicate with families easily. The entire system has been tested and the system has been working normally and shows its high performance on speech recognition and wireless transmission. The application prospect of this design is very optimistic in the field of intelligent interaction.

Paper 15:

"Hand Gesture Recognition for Deaf-Mute using Fuzzy-Neural Network" 2019 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia) 978-1-7281-3336-2/19/$31.00 ©2019 IEEE.[11]

Description: Communication is important for every individual to convey whatever information they want to people and vice versa. Hand gesture is one of the important methods of nonverbal communication for human beings. There are plenty of methods that are used to recognize hand gestures with different accuracies and precision, some has advantages and disadvantages. The general objective of this paper is to develop a hand gesture translator glove with the use of fuzzy-neural network to eliminate the barrier of communication for deaf-mute and non-deaf person. This paper studied the effectiveness of combining fuzzy logic and neural network for hand gesture recognition.

Paper 16:

Reconstruction of Convolutional Neural Network for Sign Language Recognition.

Proc. of the 2nd International Conference on Electrical, Communication and Computer Engineering (ICECCE) 12-13 June 2020, Istanbul, Turkey 978-1-7281-7116-6/20/$31.00 ©2020 IEEE.

Description: This paper presents a Sign Language translation model using Convolutional Neural Networks (CNN). A sign language is a language which allows mute and hearingimpaired people to communicate. It is a visually oriented, nonverbal communication which facilitates communication through body/facial postures, expressions and a collection of gestures. To contribute to the wellbeing of the affected population, have to implement a vision-based system to avert their day to day challenges. Our propose model constitutes object detection and classification phases. The first module is made up of single shot multi-box detector (SSD) used for hand detection. The second module constitutes convolutional neural network plus a fully connected network utilized to constructively translate the detected signs into text. The propose model is implemented using American sign language fingerspelling dataset. The propose system outperformed other published results in the comparative analysis, hence recommended for further exploitation in sign language recognition problems.

Sign language is fundamentally utilized as a communication medium between people who are dumb/deaf. Gesture based communication translation/interpretation problem is an exceptionally significant research topic as a result of its capacity to build the collaboration between the individuals who are hearing-hindered in speech. Sign language is a communication medium that uses body/facial postures, expressions and a set of gestures in human-human communication, as well as through TV and social networks. Sign Language is utilized as the first language by millions of deaf people (hearing impaired people) and people with various speaking challenges. In accordance with the investigation conducted by the British Deaf Association, it is recorded that about 151,000 people utilize Sign Language as a communication medium. There is no all-inclusive gesture based communication and pretty much every nation has its own national communication through signing and fingerspelling letters. They utilized a combination of manual gestures with facial mimics and lips articulation. These sign languages have a special grammar that has fundamental differences to speech-based spoken languages. The American Sign Language (ASL) is one of popular sign language, which has its own rules and grammar. There are sign systems such as Signed English that borrow signs from the ASL, but utilize them in English language order . As sign language involves both reading the signs (receptive skills) and rendering the signs (expressive skills), it is a twoway process. Translation and recognition of sign language is an important research area because it integrates hearing impaired people into the general public and provides equal opportunities for the entire population. A visual form communication of information between people called sign language is detailed and rapid. Both spelling and accurate translation of thoughts and feelings in a short time are very important. Since some people do not understand sign language, and some people usually find it very difficult to understand, it has become important to design a vision-based sign language translator. The design of such a system allows the communication barrier between people to be significantly reduced. There are two major methods for sign language translation. The first one is visionbased methods which utilize installed camera(s) to capture the target images which are in turn fed into the image analysis module. The second approach is a glove-based approach which utilizes sensors and gloves for implementation. Here, the additional hardware (glove) is employed to mitigate the challenges of the traditional visionbased methods. Even though signers/users often find glovebased approaches to be burdening and obtrusive, they give more accurate and precise results . This research proposes a vision-based sign language translation approach which uses a single video camera to dynamically capture hand gestures. In this paper, the proposed sign language translation (SLT) includes three basic modules: object detection, feature extraction and classification. In solving these problems, integration of three powerful models- SSD, CNN and fully connected network is proposed. These algorithms are applied for object detection, feature extraction and classification purpose. The criteria presented for the system were robustness, accuracy, high speed. The classical approach used for Sign Language recognition is fundamentally based on feature extraction and classification. In the paper, these two stages are combined in a convolutional neural network (CNN) based structure to design the sign language translation system. The presented approach simplifies the implementation of the sign language translation system. Nowadays, CNN is actively used for solving different problems. These are human activity recognition , vehicle detection in aerial images , detection of smoke as a moving object . Recently, several approaches have been presented for the purpose of sign language gesture identification. In the early 2000s, sensor-based approaches with neural networks and Bayesian networks were explored . Cheap wearable technologies such as wearable sensor gloves are utilized to get the relative gesture of hands and fingers in order to predict sign language. The use of constrained grammars and colored gloves produced low error rates on both training and test data . Using sensor devices, a multimodal framework is applied for isolated sign language translation. The sensors are used to capture finger, palm positions and then the bidirectional long short-term memory NN (BLSTM-NN) and hidden markov model (HMM) are used for classification purpose. In-depth knowledge of sign language can lead to a better understanding of the gesture classification problem. In this regard, Bheda et al. addressed the gesture classification problem using a deep CNN. In some studies, the color and depth of images are utilized for recognition purposes. Here, Ameen et al. classified ASL utilizing CNN with the color and depth of images, and in their experiments, they obtained 80% recall and 82% precision rates. The paper is organized as follows: Section two presents the proposed method utilized during design and implementation. Section three depicts simulation results and discussions. Comparative results analysis is presented in section four. Conclusively, conclusion and further thoughts are discussed in Section five.

Paper 17:

Real-Time Sign Language Recognition Based on Video Stream.

Authors: Kai Zhao1, Kejun Zhang2, Yu Zhai2, Daotong Wang2, Jianbo Su1,2

Proceedings of the 39th Chinese Control Conference July 27-29, 2020, Shenyang, China.

Description: There are millions of deaf-dumb people in the world communicating by sign language, thus designing a sign language recognition system is very meaningful and valuable for normal people to understand them. A Chinese sign language dataset is firstly created. Considering practical applications, RGB camera is used to collect video stream, instead of RGB-D camera. In order to improve the accuracy of recognition, propose a 3D-CNN method combined with optical flow processing. The collected RGB video stream is processed by optimized dense optical flow, and then put into 3D-CNN to extract feature vectors. For practical considerations, a real-time sign language recognition system is designed, composed of artificial interaction interface, motion detection module, hand and head detection module, etc. Experimental results show the superb performance and the applicability of the proposed systems.

Sign language recognition has gained a lot of attention in recent years for automatic explanations by computer or a robot. There are many application scenarios of this research, which can help deaf-dumb people communicate with others in public areas such as hospitals, banks, and train stations. The development of this research will greatly reduce the inconvenience of deaf-dumb people's lives. However, one of the problems facing sign language recognition is the lack of available sign language datasets. It has greatly prohibited the development of the practical systems. Sign language, like ordinary language, has regional differences. Different sign languages are used between different countries and even different regions of the same country. Therefore, it is hard to create a public dataset for research. Currently, the available public datasets are Chalearn14 and RWTH-PHOENIX-Weather] Chalearn14 is a dataset provided by "Chalearn Looking at People Challenge 2014", which contains 20 Italian sign language vocabularies and a total of 7,754 gesture examples. RWTH-PHOENIX-Weather is a German sign language dataset that contains 7,000 weather forecast sentences from 9 sign language speakers. Furthermore, there is so far no open and complete Chinese sign language dataset. For all Chinese researchers, creating a Chinese sign language dataset is an important task for automatic recognition. Another problem facing sign language recognition is the extraction of sign language features. The input of sign language recognition is an RGB video stream, so have to extract not only the spatial features but also the temporal features in the video stream. Traditional sign language recognition relies on artificially designed features, but when it comes to large sign language datasets, there will be great challenges. With the application of convolutional neural networks (CNNs) in computer vision, sign language recognition has also begun to extract features through 3D convolutional neural networks (3D-CNNs). Compared to 2D-CNN, 3D-CNN adds a convolution operation in the time dimension. As a feature extractor, deep 3D-CNN can extract both space and time features. Molchanov et al. propose a recurrent 3D-CNN for dynamic gesture recognition and achieve an accuracy of 83.8%. To better extract features, the authors use depth and grayscale data. However, there is currently no effective feature extraction method for RGB video streams only. In recent years, many researchers also tried to extract other features of sign language through some special equipment. Almeida et al. [4] use RGB-D sensors to obtain RGB-D video streams, which can provide RGB video stream and depth map. Each pixel value in the depth map is the actual distance of the sensor from the object. This characteristic of RGB-D video stream can be used to achieve background segmentation to better extract spatial features. Lokhande et al. Implanted an accelerator and a flexible sensor inside the glove to obtain data on the degree of bending of the fingers. However, combined with practical applications, whether it is the acquisition of depth images or the wearing of equipment, it will cause inconvenience to the daily use of deaf-dumb people. At present, the focus of sign language recognition research is on the issue of recognition accuracy. In recent years, many researchers have implemented sign language recognition based on traditional methods. Among them, Hidden Markov Model (HMM) has been widely used. Gao et al. use a self-organizing feature map and HMM method to obtain a recognition accuracy of 82.9%. Huang et al. obtained video streams from Microsoft Kinect. Multi-channels of video streams, including RGB information, depth clue, and human skeleton

positions, are used as input to the 3D-CNN in order to integrate three modes of information. Finally, on a dataset has 25 vocabularies, 88.5% recognition accuracy was obtained through the gray channel, and 94.2% recognition accuracy was obtained through the multi-channel. In order to improve the accuracy of sign language recognition without using complex feature extraction methods. propose a sign language recognition based on RGB video stream, which uses 3D-CNN combined with optical flow processing. In view of the fact that it is difficult to obtain high recognition accuracy by only using RGB video streams as input for 3D-CNN feature extraction, introduced the optical flow method. After the video stream is preprocessed by optical flow calculation, the separation of people and background can be achieved. Because real-time related issues are not considered in current sign language research, created the first real-time sign language recognition system. In order to meet the real-time requirements of real-time systems, it is necessary to reduce the time required for video stream preprocessing and feature extraction. Therefore, the frame difference method is used to reduce redundant frames in the RGB video stream,
thereby reducing the number of frames in the video stream. Motion detection, hand detection and head detection are important steps in sign language recognition systems, which can enable the system to capture RGB video in real time.Comprehensively evaluated the performance of YOLO-V3 and Faster R-CNN, and finally chose YOLOV3 to achieve hand and head detection. The reminder of this paper is organized as follows.

Paper 18:

An Efficient Hand Gesture Recognition System Based on Deep CNN.

Authors : Hung-Yuan Chung , Yao-Liang Chung , Wei-Feng Tsai.

Description: The goal of this paper is to use a webcam to instantly track the region of interest (ROI), namely, the hand region, in the image range and identify hand gestures for home appliance control (in order to create smart homes) or human computer interaction fields. Firstly use skin color detection and morphology to remove unnecessary background information from the image, and then use background subtraction to detect the ROI. Next, to avoid background influences on objects or noise affecting the ROI, use the kernelized correlation filters (KCF) algorithm to track the detected ROI. The image size of the ROI is then resized to 100x120 and then entered into the deep convolutional neural network (CNN), in order to identify multiple hand gestures. Two deep CNN architectures are developed in this study that are modified from AlexNet and VGGNet, respectively. Then, the above process of tracking and recognition is repeated to achieve an instant effect, and the system's execution continues until the hand leaves the camera range. Finally, the training data set can reach a recognition rate of 99.90%, and the test data set has a recognition rate of 95.61%, which represents the feasibility of the practical application.

The process of hand gesture recognition generally has two parts: detection and recognition, wherein the recognition is also divided into dynamic hand gestures and static hand gestures. The static hand gesture means a fixed hand gesture, and the dynamic hand gesture refers to continuous motion recognition such as waving and grabbing. First, for detection, the background and the hand are generally segmented using the skin segmentation method, and then the noise processing is performed and the background subtraction method is used to obtain the desired region of interest (ROI), namely, the hand region. In recent years, because of the Kinect depth camera introduced by Microsoft, many depth information-based methods have emerged, such as Keskin et al. and Memo et al. , which use the random forest , a machine learning method, to train the model to capture the hand's skeletal structure. However, because the price of such a camera is relatively expensive compared to a typical web camera and tends to be affected by the light source of the location, there are still many limitations in terms of application. As for the identification aspect, its essence is classification. Classification is performed by setting different hand gestures into different categories, using manually set decision criteria (traditionally) or trained classification models (in recent years). The traditional method is to perform the recognition by using the convex hull of the hand after performing skin segmentation. Recognition is determined by the number of polygon edges generated by a hand gesture, and only the numbers from 1 to 5 can be recognized. For example , controlling the robot arm with this method can make very little variations, and it is susceptible to complex background interference. Moreover, the hand must be completely facing the lens and this cannot be done on some complicated hand gestures. In recent years, many research teams have adopted machine-learning methods to train models for classification, such as support vector machine , hidden markov model , convolutional neural network (CNN) , recurrent neural network and so on. Among them, CNN is more popular in the field of recognition, and has better results than other methods, mainly because it can get the required feature values from the input picture, and can learn the difference between different samples well by using a large number of samples in its training. However, in the past, its development has been limited due to the speed of hardware computing. In recent years, due to the advancement of semiconductor manufacturing, the computing speed of graphics processing units is getting faster, and the bottleneck of hardware processing speed has been addressed, allowing the CNN network to develop rapidly to become the deep CNN network.

Paper 19:

Gesture detection from RGB hand image using modified convolutional neural network.

Authors : Zhihua Hu , Xiaoming Zhu.

2019 2nd International Conference on Information Systems and Computer Aided Education (ICISCAE) 978-1-7281-3066-8/20 ©2020 IEEE 10.1109/ICISCAE48440.2019.221606

Description: Study the RGB-only hand gesture recognition in human computer interaction. First, the hand region and landmarks are detected, in order to remove the complex background and convert the RGB image to iconic image. Active Shape Model is adopted for landmark detection and Markov Random Fields are used for error correction. Second, propose to use a modified CNN model to recognize the visual hand gesture information. Two stages are tested in experiments and compared with conventional machine learning models. Experimental results show that our proposed method has an advantage in recognition accuracy.

Hand gesture recognition (HGR) is an important research topic in intelligent human-computer interaction (HCI) . In human computer interaction applications, speech, language, facial expression, hand gestures, etc., are the common channels for machines to understand user intention and for human to express the intention. Children's education related to intelligent robots is increasingly valued by parents. Gesture recognition adds a new way of human-computer interaction to robot education, adding children's fun during learning. Detecting hand gestures can be very helpful to improve user experience in a typical HCI system. Understanding the hand gestures from an RGB only image is challenging. Various background objects and illumination conditions may interfere the detection of hand area. Hand and gesture recognition have draw many research attentions. Lu et. al, proposed to apply gesture recognition in mobile devices to improve the natural interaction between human and smart cellphone. They used accelerometer sensors to implement the gesture recognition algorithm. F. Liu et. al, used kinect depth information to improve the hand gesture recognition performance. K. Liu et. al, proposed a vision-based real-time hand recognition framework based on cameras. However, their approach requires stereo images that are not easy to achieve on common consumer mobile devices. Ju et. Al proposed to use RGB-D image to recognize different gestures, however the alignment in such systems is usually problematic. Previous efforts made on hand gesture recognition mainly concern the RGB-D (RGB image and depth image)approach. The depth channel provides useful information to separate the foreground object and the background object. When the HGR interaction scene is simple under controlled condition, may take the foreground object as the detected hand.

However in many real world applications, such as mobile devices, a depth image is not available.

Paper 20:

Gesture Recognition based on Deep Convolutional Neural Network.

Authors : P. Jayanthi, Dr. Ponsy R. K. Sathia Bhama

Description: A sign language is visually conveyed signal patterns to pass on meaning by concurrently coalescing hand shapes and direction and arms movement, body movement , facial expressions to articulate gracefully a narrator's feelings. Movement of the hand called hand gesture is the part of sign used by hearing-impaired inhabitants. It plays a key role for finger spelling in which signers spell out a word as a chain of hand shapes or hand flight equivalent to individual letters. Hand gesture recognition research classified as Glove based and vision based; Glove based system affects the natural signing ability. Vision based category is not using any devices or gloves, is the most natural way. Data acquisition can be done with camera. Conventional method of gesture recognition is time consuming so the usage of deep CNN is to prevent the segmentation or detection stage which favors faster real-time process.

Approximately 70 million Deaf people around the globe converse by means of sign language as different from verbal language. For communication, the Sign Language uses facial and body movements. Like many spoken languages Sign language is not a common language, and diverse sign languages are used in different countries. Quite a few nations as India, the USA and the UK have more than one sign language. Hundreds of sign languages are there in existence like American Sign Language (ASL), British Sign Language (BSL), Indian Sign Language (ISL), Japanese Sign Language, (JSL), Turkish Sign Language (TID), , Chinese Sign language (CSL), Arabic Sign Language (ArSL), Spanish Sign Language (LSE), Australian Sign Language (Auslan), Chilean Sign Language (LSCh) etc. Over 21 sign languages are officially recognized by constitution and laws of states, for instance ASL, Auslan, BSL, LSCh, etc. Many sign languages are not officially recognized. Sign languages are natural languages as other (spoken) languages because they are controlled by the brain's left hemisphere. Sign language has acknowledged for its existence from fifth century BC. In 17th century Zuan Pablo Bonet in Madrid, Spain did the earliest footage of work on sign language, who published a manuscript entitled Reducción de lasletras y arte para enseñar a hablar a los mudos ('Simplification of Sounds and Art for Teaching Deaf to Speak'), 1620. The signs made use in his book are depicted in Sign language signs can be static (motionless/ posture) or dynamic (lively/gesture). Father of ASL linguistics, polyglot William Stokoe and the ASL glossary , a sign is expressed in terms of four elements: hand shape, hand movement, palm orientation and location in relation to the body. Static sign is derived from the fingers position with respect to palm i.e., Hand shape and Dynamic signs are extracted from static sign, along with the palm orientation. Formulating algorithms and systems to discover a series of formed signs and to recognize their meaning in the form of content or verbal communication is referred to as sign language recognition. Sign language recognition (SLR) follows a line of investigation involving artificial neural networks, computer vision and linguistics and etc. Since the signs are done through hands include both single hand and two hands, facial expressions include eyes and eyebrows and other parts of the body and it is a wide-ranging issue. A sign language recognition (SLR) system is categorized into two groups : vision-based systems and glove based systems. The later faction necessitates end user to put on gloves. The glove facilitates the recognition to evade or make simpler the segmentation and detection. On the other hand, users have to bear a hardware gadget that formulates them. Besides, the glove based System possibly mislay the emotional expression of the signer from facial expression, essential for sign language recognition. On the other hand, the earlier faction, rely on vision based techniques where the sequence of signs are captured with video camera, is easier for users. Nevertheless, difficulty is how to do precisely segmentation and tracking.

Paper 21:

Human Action Recognition using Deep Neural Network.

Authors: Rashmi R. Koli,Tanveer I. Bagban.

2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability 978-1-7281-6823-4/20 2020 IEEE

Description: Human activities such as body gestures are the most difficult and challenging things in deep neural networks. Human action recognition is nothing but the human gesture recognition. Gesture shows a movement of body parts that convey some meaningful message. Gestures are greater most suitable and natural to have interaction with systems(computer) for humans thus it builds a platform between humans and machines. Human activity recognition provides the platform to interact with the deaf and dumb person. In this research work, introduce to develop a platform for hand movement recognition, which recognizes hand movement(gestures), by using the CNN, can identify human gestures in the image. As it aware there was a quick increment in the amount of deaf and dumb victims because of a few conditions. Since deaf and dumb people can't communicate with a typical person so they want to depend upon a sort of visual correspondence Sign language. The sign language gives the fine verbal exchange platform for listening to impaired men or women to bring their minds and to have interaction with a normal character. The purpose of this research is a gadget that broadens the popularity of gestures, which can recognize gestures and then convert gesture images into text accordingly. The system pays special attention to the CNN training component using the CNN algorithm. The concept includes designing a gadget that uses in-depth mastery standards to treat input as a gesture and then provide recognizable output as text.

Human action recognition is an important aspect of humanto- human communication as it provides information about human nature. For example, the identity of a person, their personality, and the psychological state, it is hard to extract. Today, for some reason, the number of deaf-mute people is increasing rapidly. It recognizes that deaf and dumb people cannot talk to normal people, while ordinary people cannot understand the meaning of gestures. Human action popularity affords the pleasant verbal exchange platform for these human beings to engage with normal humans. In this system, it cognizances on a vision-primarily based hand reputation method which is greater natural and comfortable for information the hand gestures. There are many feature extraction strategies, and there is also classification strategies. Choosing which strategy to use is a very difficult task. The most important and crucial method is segmentation, in segmentation foreground is separated from the background. This separation needs feature extraction methods for calculation, such as angle calculation, calculate accuracy as well as the result prediction. In this video action recognition, every hand movement is an have different meaningful textual form. This method is slightly different from still images, because human action is consisting of everchanging motions or collection of moving elements. So that it is important to find which gesture is appropriate to which meaningful text. In this way, it's essential to investigate differing spatial-temporal highlights for activity acknowledgment. Using deep learning it makes use of motion information in various ways. Neural networks can identify automatically learning features from large amounts of datasets. Using some algorithms CNN can handle each frame information is correct and appropriately result. The main intention of this research is to make a system for deaf and dumb people's hand gestures recognizer, which recognizes the gesture and converted into proper meaningful text.

Paper 22:

Hand Gesture Recognition using Convolutional Neural Networks.

Authors: Shengchang Lan, Zonglong He, Weichu Chen, Lijia Chen.

978-1-5386-7105-4/18/$31.00 2018 IEEE

Description: This paper introduced a hand gesture recognition method based on convolutional neural networks (CNNs). The recognition scenario consisted in a three dimensional radar array to transmit and receive 24GHz continuous electromagnetic (EM) wave, and convert the scattered EM wave to the intermediate frequency (IF) signals. This paper used the the processed frequency spectrum as the input to the CNN. Then the CNN feature detection layer learned through data training, avoiding supervised feature extraction while learning implicitly from training data. It highlighted these features through convolution operating, pooling and a softmax function. Results showed that this system could achieve a high recognition accuracy rate higher than 96%.

With its success in the fields of image recognition and classification, convolutional neural networks (CNNs) has attracted great attention in many fields . In 1960s when Huber and Wiesel studied neurons for locality sensitivity and direction selection in the cortex of the cat, they discovered its unique network structure can effectively reduce the complexity of feedback neural network[3]. Currently, CNNs have become one of the research hotspots in many fields, especially in the field of pattern classification. The network has been widely used because it avoids the complicated pre-processing of images and can directly input original images. In recent years, developing more efficient human-computer interaction becomes increasingly attractive in the computer science society, aiming to digitally interpreting the hand gesture information to the computers based on visual images and wearable inertial sensors. However, microwave engineering has involved in this new realm as an emerging solution. Featured for simple structure, high sensitivity and narrow spectrum occupancy, Frequency Modulated Continuous Wave (FMCW) radar is able to detect the frequency shift when the transmitted electromagnetic wave is scattered by the moving targets according to Doppler Effect. In this paper, it is investigated the feasibility of designing a hand gesture recognition system using convolutional neural networks to classify basic human hand gestures including hand motion, finger motion and fist motions. The experiment result showed a satisfactory accuracy rate higher than 96% in the recognition and verified the proposed method with a promising vision in controlling the modern electronics.

Paper 23:

Video Recognition of American Sign Language Using Two-Stream Convolution Neural Networks.

Authors: Fikri Nugraha, Esmeralda C. Djamal.

2019 International Conference on Electrical Engineering and Informatics (ICEEI) July 2019, 9 - 10, Bandung, Indonesia 978-1-7281-2418-6/19/$31.00 ©2019 IEEE.

Description: Sign language uses manual-visual to convey meaning. The style is expressed through manual sign flow in combination with non-manual elements. Sign gestures interpreted in the meaning of words, letters, and numbers. This study proposed Two-stream Convolutional Neural Networks (CNN) to recognize and classify words in hand motion images of video form. Two-stream CNN works with two processes, namely spatial and temporal stream. Spatial flow detects edges and overall global features. While temporal flow identifies local action features in stacked optical flow images of 10 frames, each stream passed Softmax function. Average Fusion function combines both of streams. Two-stream separated training reduced computing time and overcome resource limitations. In building a CNN two-stream model, a specific configuration is needed to update the weight during training such as VGG – SGD, Resnet – Adam, Resnet – SGD, Xceptionnet – Adam, and Xceptionnet – SGD. The result gave the best precision used Xceptionnet SGD of spatial flow and Xceptionnet Adam of temporal flow configuration. The architecture gave precision 89.4% of a combination of one choice or Top1 is 89.4% and 99.4% of the five choices or Top5.

Sign language is communication that involves body language and lip movements without using sound. Usually, the sign language used meaning communication for hearing impaired and deaf people who have limited oral communication. Therefore, like language in general, sign language also uses standard provisions that are easy for everyone to understand. Usually, hearing impaired and deaf people are accustomed to memorizing and understanding hand movements as sign language to communicate with their fellow people as this is the most effective communication needs. However, the problem arises if the deaf person wants to communicate with the ordinary person. They cannot understand the sign language. For people who can communicate orally, the use of sign language is not a concern, so it is not easy to learn it. Sign language is sawed as a sequential image that has the characteristics of specific symbols. The rapid development of machine learning, such as Deep Learning allows the system to learn and remember patterns of hand movements and facial expressions as certain symbols in sign language. Some researches recognized gesture movement patterns using Deep Learning to identify human hand objects using the Convolutional Neural Network (CNN). The kernel or filter used in the convolution chosen randomly. Convolution results in a layer affect the selection of the kernel in the next layer. The classification layer in the study used fully connected Artificial Neural Networks . The use of the Graphical Processing Unit (GPU) devices that are accompanied by Tensor as a graphics processing function can increase computing speed . Tensor is a multi-linear function that contains data structures to do alternative computations for computing arrays or vectors. Other studies recognized the Indonesian language sign system (SIBI) in the hand part by manually segmenting the image captured through the Kinect camera . The study only recognized letters and accuracy figures, with 86% with grayscale images and 83% with depth images. Real-time sequential image recognition has been done to identify letter and number sign languages using Two-Stream CNN . However, both studies only focus on the introduction of letters and numbers, whereas sign language usually uses words more like symbols. Moreover, the problem lied in computational time for training up to a day. Other studies have recognized American sign language using the Hierarchical Attention Network with Latent Space for subsequent image processing, whereas CNN is used to identify words for each frame . The model has extensive feed-forward dense layer attention network so required the computational time of around 12 hours for training. Moreover, the model needs third-party devices such as Kinect for obtaining video datasets. Therefore, this research will have a problem if real-time implementation. Motion video processing, which is a sequential image, has been improved in other studies using FlowNet, namely by estimating the displacement distance between subsequent frames. The research used five frames every second but is limited to moving images that have little difference between frames. The use of GPU graphics processing devices produces better computing speed. Then various motion recognition scenarios without limiting image differences between frames with an accuracy of 79%. However, there are obstacles in computing time. The study used Chair Datasets and Thing3D Datasets Flownet. Meanwhile, sign language is a sequential image, so each frame of movement video require specific image processing to represent a symbol. Because the case so complicated so requires more than just CNN each frame. Therefore need a method to combine image analysis of each frame using Two-stream CNN. The process extracts the features of each frame into a five-frame image series as input from the classification layer on CNN . While other studies recognized the action of hand movements by adding dimensions to the CNN, each time unit represented by the frame becomes a representation of the dimensions of height, width, depth and time so that convolution can be carried out on multiple images or multiple frames at once. The research also brings the consequences of computing becoming more complex because the number of feature parameters obtained becomes computationally very expensive. This study proposed a Two-stream CNN method for identifying words from American sign language videos. The two processes used are spatial and temporal and trained separately. This method is intended to minimize resource limitations due to computational time problems . The Two-stream CNN model works with a specific configuration to update the weight during training such as VGG – SGD, Resnet – Adam, Resnet – SGD, Xceptionnet – Adam, and Xceptionnet – SGD.

Paper 24:

Hand Gesture Recognition and Implementation for Disables using CNN'S.

Authors: Kollipara Sai Varun, I. Puneeth and T. Prem Jacob.

International Conference on Communication and Signal Processing 2019 India.

Description: Hand Gestures are playing the major role in today's industry. The development of the gesture recognition or detection will help people in different ways. Development of many automated technologies like machine learning, deep learning, neural networks and computer vision will help in using futuristic methods like gesture controlling and recognition. These gestures can be used in such a way it can help with people who have difficulty in controlling or operation systems or devices. Here the development of the model is in such a way it can help us to recognition and implementation. For such developments these models will be working with Convolution Neural Networks and Back Propagation methodologies which can help for the easy working of these models.

The communication or transfer of data in between human and human is really easy and understandable. But when it comes to human and machine it's really difficult because even machine knows all the languages humans can speak and understand, they cannot communicate with that knowledge or data. So for improving that communication features with machines it can develop some interactive techniques like gesture recognition [1]. This is becoming a trending topic in the field of computer science and its applications related to deep learning technology [2]. Hand Gesture acknowledgment is essential for structuring no touch or control interfaces in vehicles. Such technologies enable drivers to drive while at that time connecting with different controls, e.g., sound and cooling, and also there are line enhance drivers' security and solace. In the recent decades, numerous vision-based powerful hand signal acknowledgment [3-5] calculations were presented. To perceive motions, distinctive highlights, for example, handmade spatiotemporal descriptors and enunciated models were utilized. As signal classifiers, concealed Markov models, contingent irregular fields and bolster vector machines (SVM) have been broadly utilized. Notwithstanding, vigorous order of signals under broadly fluctuating lighting conditions, and from various subjects is as yet a testing issue. Computer-human communication refers to the way how the human communicate to the computer/machine, and since the machine is not useful until a human trains the machine for a particular task. There are mainly 2 characteristics that

will be checked when developing a man-machine communication model as mentioned in: machine's performance and usage. The Model performance refers to how well the machines are
performing to communicate with the human and usage refers to weather all the provided functionalities are performing according to the development. Gestures can be in any form like hand image or pixel image or any human given pose that require less computational difficulty or power for making the devices required for the recognitions to make work. Different techniques are being proposed by the companies for gaining necessary information/data for recognition handmade gestures recognition models . Some models work with special devices such as data glove devices and color caps to develop a complex information about gesture provided by the user/human. Recently, visual communications like augmented reality and virtual reality has attracted attention since many companies have introduced futuristic devices for the company's growth. For example, smart glasses for realistic features and many applications that have been developed in many companies. Virtual glass and augmented devices displays have also been introduced and are being used in futuristic applications. Hand gesture recognition is widely used as an interface to give commands to these reality controlled devices. The system that acquires a perfect balance between the concepts considered as well performed and super powerful systems. Gestures can acts as communication barrier between the human and machines. Also can develop between people who cannot communicate properly.

Paper 25:

Joint Hand Detection and Rotation Estimation Using CNN.

Authors: Xiaoming Deng , Yinda Zhang, Shuo Yang, Ping Tan, Liang Chang, Ye Yuan, and Hongan Wang.

IEEE Transaction On Image Processing , VOL. 27, NO. 4, APRIL 2018.

Description: Hand detection is essential for many hand related tasks, e.g., recovering hand pose and understanding gesture. However, hand detection in uncontrolled environments is challenging due to the flexibility of wrist joint and cluttered background. It proposes a convolutional neural network (CNN), which formulates in-plane rotation explicitly to solve hand detection and rotation estimation jointly. Our network architecture adopts the backbone of faster R-CNN to generate rectangular region proposals and extract local features. The rotation network takes the feature as input and estimates an in-plane rotation which manages to align the hand, if any in the proposal, to the upward direction. A derotation layer is then designed to explicitly rotate the local spatial feature map according to the rotation network and feed aligned feature map for detection. Experiments show that our method outperforms the state-of-theart detection models on widely-used benchmarks, such as Oxford and Egohands database. Further analysis show that rotation estimation and classification can mutually benefit each other.

Locating human hands and knowing their pose are extremely useful in human-computer interaction and robotics. It helps computers and robots to understand human intentions , and provides a variety of clues, e.g. force, pose, intention, for high level tasks. While generic object detection benchmarks have been refreshing over the last decade, hand detection and pose estimation from a single image, however, is still challenging due to the fact that hand shapes are of great appearance variation under different wrist rotations and articulations of fingers . In this paper, it propose to solve the hand detection problem jointly with in-plane rotation estimation. These two tasks are closely related and could benefit each other. Calibrating training data under different rotation to upright position results in rotation invariant feature, which relieves the burden of the detection/classification model. While in return, detection model can verify if the rotation estimation is reasonable. However, due to the nature of the convolutional neural networks, rotation invariance is more difficult to achieve than translation invariance, which prevents us from an end-to-end optimization. As a result, previous work usually handle transformation estimation and detection separately or in an iterative fashion, which may not achieve a good optima. To tackle this issue, it is designed to derotation layer, which explicitly rotates a feature map up to a given angle. This allows us to jointly optimize the network for two tasks simultaneously. Recently, spatial transformer networks (ST-CNN) also presented a differentiable module to actively spatially transform feature maps with CNN. However, their transformation is learned unsupervised such that could be any arbitrary rotation that can not be interpreted directly (the discussion that ST-CNN may not be an ideal hand detection model are shown in the appendix). Also, the transformation space is typically huge and would require much more data and time to converge. Comparatively, our rotation estimation network is aimed for upright alignment, such that the output can be directly used for related tasks, e.g. hand pose estimation. It is also trained supervised, which is more likely to converge.Our network is built on the backbone of Faster R-CNN with a derotation layer inserted after the ROI pooling. Taking the local feature after ROI pooling for each of the proposal as input, the rotation estimation network outputs an in-plane rotation, which is taken by the derotation layer to align the local feature map. The detection network then takes the aligned feature map and produces a binary classification to tell if the proposal contains a hand. The overall model can be trained end-to-end and runs efficiently during testing process. The contributions of this paper are mainly in two aspects. First, it propose, by our knowledge, the first framework that jointly estimates the in-plane hand rotation and performs detection. Experiment shows that it achieve significant better performance than state-of-the-art on widely used benchmarks.

3. ANALYSIS

3.1 Detailed Statement of the problem

The Problem of Deaf and Dumb people is they cannot speak like a normal people. So, they find it bit difficult to tell their needs and communicate to normal people. Our system uses machine learning model to predict sign language done by deaf and dumb people. The primary objective of our system is to predict the sign and give output of their sign in textual format.

Assistive technology has made a significant impact on the lives of people with disabilities, providing them with greater independence, social inclusion, and improved quality of life. In the case of the deaf and dumb community, real-time assistive systems can play a vital role in enhancing communication and accessibility.

One such system that has gained popularity in recent years is the use of computer vision and machine learning models to recognize sign language gestures in real-time. YOLOv5 (You Only Look Once version 5) is a state-of-the-art machine learning model that can be used to develop such a system. YOLOv5 is an object detection algorithm that is capable of detecting and recognizing multiple objects within an image or video frame with high accuracy and speed.

The real-time assistive system for the deaf and dumb community using YOLOv5 would consist of a camera that captures the sign language gestures made by the user. The captured video frames would then be fed into the YOLOv5 algorithm, which would detect and recognize the sign language gestures in real-time. The recognized gestures would then be converted into text or audio, enabling communication with non-sign language users.

To develop such a system, several steps are involved. The first step is to collect a large dataset of sign language gestures performed by different users. This dataset would be used to train the YOLOv5 model to recognize the different sign language gestures accurately. The dataset would need to be diverse, containing examples of various sign language alphabets, words, and sentences.

Once the dataset is ready, the next step would be to train the YOLOv5 model using the collected data. During training, the YOLOv5 model would learn to recognize the different sign language gestures by identifying key features and patterns within the dataset. The model would need to be trained using a powerful graphics processing unit (GPU) to reduce the training time.

After training, the YOLOv5 model would be integrated into the real-time assistive system for the deaf and dumb community. The camera would capture the sign language gestures made by the user, and the captured video frames would be fed into the YOLOv5 model. The model would detect and recognize the sign language gestures in real-time and convert them into text or audio.

The final step would be to test and evaluate the real-time assistive system's performance using a diverse range of sign language gestures and different users. The system's accuracy, speed, and user-friendliness would be evaluated to ensure that it is effective and efficient in assisting the deaf and dumb community in communicating with non-sign language users.

In conclusion, the real-time assistive system for the deaf and dumb community using YOLOv5 is a promising technology that has the potential to enhance communication and accessibility for people with disabilities. By leveraging machine learning models, such as YOLOv5, we can develop systems that are accurate, fast, and user-friendly, ultimately improving the quality of life for people with disabilities.

3.2 Requirement Specifications

A Software requirements specification (SRS) is a detailed description of a software system to be developed with its functional and non-functional requirements. The SRS is developed based the agreement between customer and contractors. It may include the use cases of how user is going to interact with software system. The software requirement specification document consistent of all necessary requirements required for project development. To develop the software system, we should have clear understanding of Software system. To achieve this, we need to continuous communication with customers to gather all requirements.

Requirements gathered for any new system to be developed should exhibit the following three properties:

a) Unambiguity

b) Consistency

c) Completeness

Requirements can be specified in terms of Hardware and Software requirements as follows:

Hardware Requirements (minimum)

1) Computer System or Laptop with:

- OS: Windows 7 and above
- RAM: 8 GB
- HardDisk:500 GB

Software Requirements

- PyCharm
- TensorFlow
- Visual Studio 2019
- YOLOv5 Open-Source ML Model

3.3 Functional Requirements

Functional requirement defines a function of a system or its component. A function is inscribed as a set of inputs, the behaviors, and outputs. Basically, requirements are statements that indicate what a system needs to provide capability.

- We must be able to trace static frames from the live data properly.
- We must be able to Extract the labelled sign for the desired sign for user side.

3.4 Non-Functional Requirement

The Non-functional requirement elaborates a performance characteristic of thesystem.

Non-functional requirements of this project are:

1. Accessibility: The application is easily accessible.
2. Availability: The application is available and accessible anytime.
3. Recoverability: The application is easily recovered.
4. Maintainability: The application is easy to be understood and maintain.
5. Usability: Interactive User Interface.
6. Efficiency: The application is very efficient.
7. Robustness: The application is strong.
8. Reliability: The application is highly reliable.

3.5 Feasibility Study

A feasibility analysis usually involves a thorough assessment of the operational (need), financial and technical aspects of a proposal. Feasibility study is the test of the system proposal made to identify whether the user needs may be satisfied using the current software and hardware technologies, whether the system will be cost effective from a business point of view and whether it can be developed with the given budgetary constraints. A feasibility study should be relatively cheap and done at the earliest possible time. Depending on the study, the decision is made whether to go ahead with a more detailed analysis. When a new project is proposed, it normally goes through feasibility assessment. Feasibility study is carried out to determine whether the proposed system is possible to develop with available resources and what should be the cost consideration. Facts considered in the feasibility analysis were-

- Technical Feasibility.
- Economic Feasibility.
- Behavioral Feasibility.

a) Technical Feasibility

Technical feasibility includes whether the technology is available in the market for development and its availability. The assessment of technical feasibility must be based on an outline design of system requirements in terms of input, output, files, programs and procedures. This can be qualified in terms of volumes of data, trends, frequency of updating, cycles of activity etc., in order to give an introductionof technical system.

b) Economic Feasibility

The economic feasibility study represents tangible and intangible benefits from the project by comparing the development and operational cost. The technique of cost benefit analysis is often used as a basis for assessing economic feasibility. This system needs some more initial investment than the existing system, but it can be justifiable that it will improve quality of service. Thus, feasibility study should center along the following points:

- Improvement resulting over the existing method in terms of accuracy,timeliness.
- Cost comparison.
- Estimate on the life expectancy of the hardware.

c) Behavioral/ Operational Feasibility

This analysis involves how it will work when it is installed and the assessment of political and managerial environment in which it is implemented. People are inherently resistant to change, and computers have been known to facilitate change. The new proposed system is very much useful to the users and therefore it will accept broad audience from around the world.

3.6 Use Case Diagrams

To model a system the most important aspect is to capture the dynamic behaviors. To clarify a bit in detail, dynamic behaviors mean the behaviors of the system when it is running or operating. So only static behaviors are not sufficient to model a system rather dynamic behaviors are more important than static behaviors. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we must discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. So, use case diagrams are consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system. So, to model the entire system numbers of use case diagrams are used.

The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and State chart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements.

Purpose of Use Case Diagram

Use case diagrams are typically developed in the early stage of development and people often apply use case modeling for the following purposes:

- Specify the context of a system.
- Capture the requirements of a system.
- Validate a systems architecture.
- Drive implementation and generate test cases.
- Developed by analysts together with domain experts.

Use cases share different kinds of relationships. Defining the relationship between two use cases is the decision of the software analysts of the use case diagram. A relationship between two use cases is basically modeling the dependency between the

two use cases. The reuse of an existing use case by using different types of relationships reduces the overall effort required in developing a system.
Use case granularity refers to the way in which information is organized within use case specifications, and to some extent, the level of detail at which they are written. Achieving the right level of use case granularity eases communication between stakeholders and developers and improves project planning.

Use-case diagrams are helpful in the following situations:

- Before starting a project, you can create use-case diagrams to model a business so that all participants in the project share an understanding of the workers, customers, and activities of the business.
- While gathering requirements, you can create use-case diagrams to capture the system requirements and to present to others what the system should do.
- During the analysis and design phases, you can use the use cases and actors from your use-case diagrams to identify the classes that the system requires.
- During the testing phase, you can use use-case diagrams to identify tests for the system.

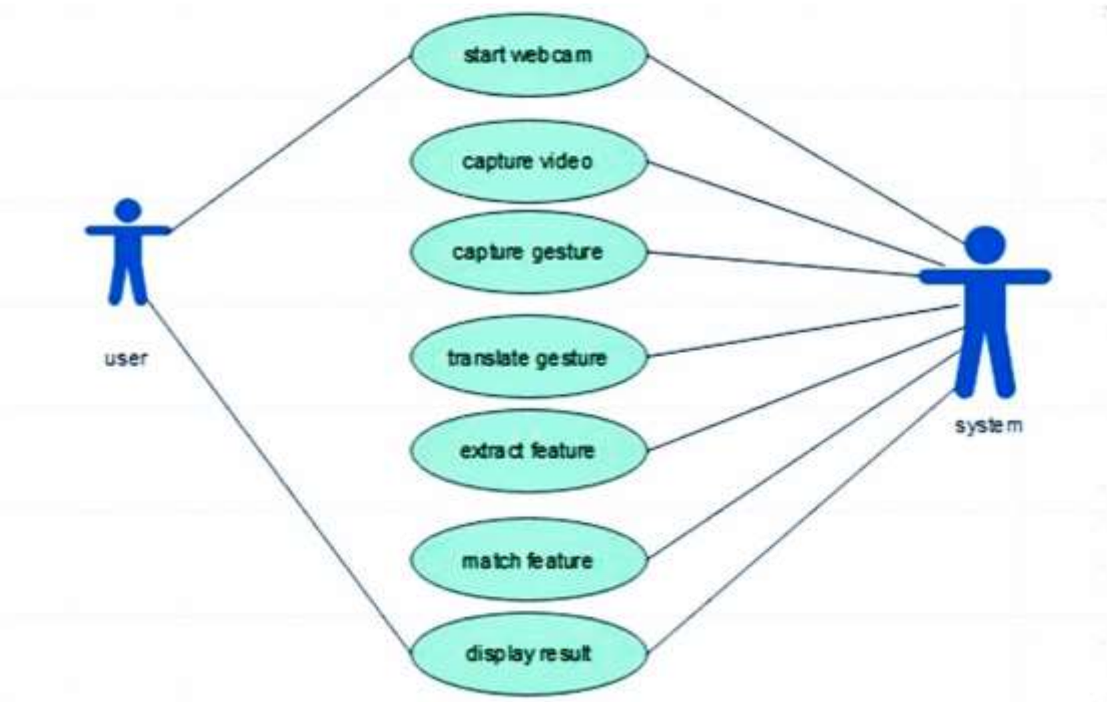


Figure 3.6.1: Use Case Diagram

3.7 Use Case Specification

A Use Case Specification is a textual description of the functionality provided by the system. It captures actor-system interaction. That is, it specifies how a user interacts with a system and how the system responds to the user actions. It is often phrased in the form of a dialog between the actor and the system. The use case specification is represented in the use case diagram by an oval and is what most people think of when they hear the term use case.

A Use Case Realization describes how a use case, which is logically defined by the use case specification, is physically implemented within a design model in terms of collaborating objects. It organizes the design model artifacts related to the use case. It often comprises multiple design artifacts such as a class diagram, object diagram, sequence diagram, etc. that describe how the physical design will implement the use case specification.

The purpose of use case realization is to separate the concerns of the system stakeholders, which are typically captured by the use case model and system requirements, from the concerns of the system designers. In doing so, the designers can choose to implement the use case specification without affecting the use case specification.

Depending on what you are interested in, you would begin reading with an actor or with a business use case. Starting with the actor, passenger, we find the associations (lines) to the two business use cases, check-in and express check-in . This means that people, who appear as passengers, can either go through check-in, or express check-in, which can be conducted without luggage.

That one of the two business use cases is below the other means nothing. A use case diagram does not document a meaningful order in which business use cases could be conducted. Of course, the order matters for the description and linking of business processes.

The actor check-in representative also has an association to the business use case check-in This means that not only the passenger, but also someone who represents him or her can check in. That the actor, passenger, also has an association to the use case check-in means that the passenger and the check-in representative can both check-in. However, what the diagram does not show clearly is that it does not mean that they perform the check-in together. This fact can only be described in another diagram or in the form of a comment that can contain informal text.

That the actor check-in representative only has an association to the business use case check-in means that at the UML Airport a representative of the passenger cannot perform an express check-in

You can see that such a simple diagram can contain quite a lot of information. The business use case check-in and the business use case express check-in each have an include relationship with issuing boarding pass Use case diagrams. Both use the business use case issuing boarding pass at some point in their own interaction. (Use cases cannot define when another use case is executed.) Sometime during check-in, the boarding pass is issued and handed to the passenger or check-in representative.

As a first step, it is important to find knowledge carriers, for analysts and knowledge carriers to work out the basic principles together. Such knowledge carriers are, for example:

- People who are involved in performing, operating, and controlling business processes.
- Users of similar or related IT systems.
- Customers, who are often critical and creative knowledge carriers.
- Business partners.
- Domain experts.
- Management.
- External observers.

Several helpful techniques have proven to be practical for the analysis and understanding of business processes:

- Observing employees at work.
- Participating in the business processes being investigated.
- Taking the role of an outsider (e.g., of a customer).
- Giving out surveys.
- Performing interviews.
- Brainstorming with everyone involved.
- Discussing with domain experts.

- Reviewing existing forms, documentation, specifications, handbooks, and work tools.
- Describing organizational structure and workflow management.
- Reviewing organization charts and job descriptions.

Actors:

User:

- User can start camera.
- User can view result.

Use Cases:

- Start webcam.
- Capture Video.
- Capture Gesture.
- Translate Gesture.
- Extract Feature.
- Match Feature.
- Display Result.

4. DESIGN

4.1 Design Goals

Design is a meaningful engineering representation of something that is to be built. It can be traced to a customer's requirements and at the same time assessed for quality against a set of predefined criteria for good design. In the software engineering context, design focuses on four major areas of concern: data, architecture, interfaces, and components. The design process translate requirement into representation of software that can be accessed for a quality before core generation. Design is the process through which requirement are translated to blue print for constructing into software. Initially the blueprint depicts the holistic view of software. This is the design represented at the high level of abstraction.

During various stages of system development and design following goals have been setup for a complete architecture

- Analysis
- Design
- Development
- Testing
- Implementation

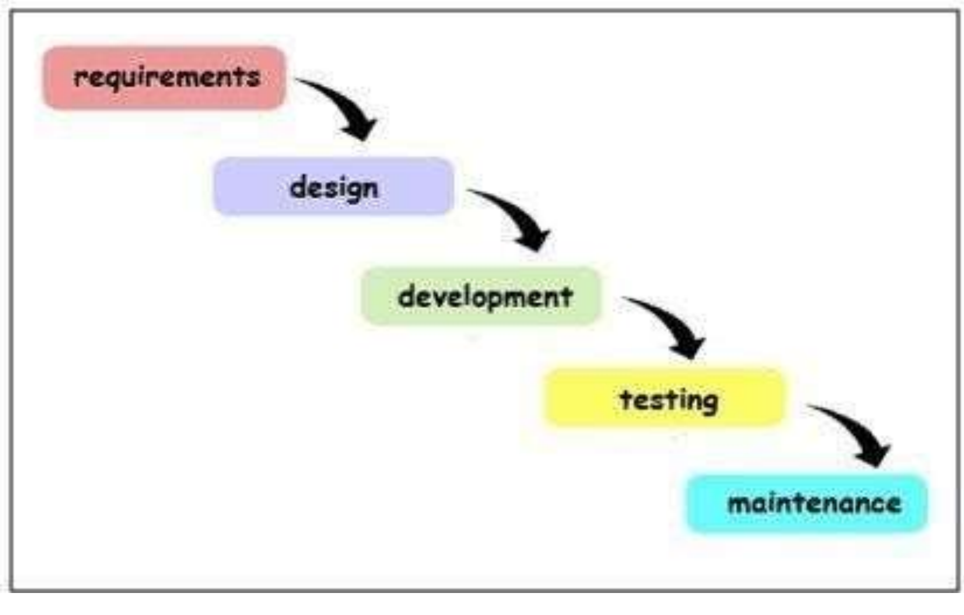


Figure 4.1.1: SDLC Cycle

The design goals of our project are quite simple. The customer satisfaction is our priority. We try to keep the user interface simple and clean.

The waterfall model depicts the software development process in a linear sequential flow; due to this, it is also referred to as a linear-sequential life cycle model, which indicates that any development process steps can start only after the previous one has finished. The stages are always done in this order and never overlap. Before moving on to the next step, the developer must finish the present one. The model is called a waterfall because it progresses from one phase to the next logically.

There is an emphasis on the natural succession of these phases. During the SDLC phase, each step is meant to execute specific tasks.

This model can be used in the scenarios were,

- The requirements do not often change, and the clients have a crystal-clear understanding about what they want as the output from the software they wanted to be developed because there would be no scope of changing requirements in future.
- You are working on an application that is not complicated, for example, standalone applications which are short, like Customer Relationship Management (CRM) systems, Human Resource Management Systems (HRMS), Supply Chain Management Systems, etc.
- You have the clarity of the requirements provided by the clients, to achieve this step you may have query sessions with the users and can discuss the dos and don’ts of the software.
- The technology and tools used in the application are not dynamic.

Waterfall Methodology can be used when:

- Requirements are not changing frequently.
- Application is not complicated and big.
- Project is short.
- Requirement is clear.
- Environment is stable.
- Technology and tools used are not dynamic and is stable.
- Resources are available and trained.

Phases of Waterfall Model in Software Engineering:

There are several phases in the waterfall model. They are briefly explained below. Let us understand the concept of Waterfall model with example of a banking application for illustrating the topic.

Let us assume that the Citibank is planning to have a new banking application developed and they have approached your organization in the 1990's.

Requirements Gathering and Analysis:

In this phase the requirements are gathered by the business analyst and they are analyzed by the team. Requirements are documented during this phase and clarifications can be sought.

The Business Analysts document the requirement based on their discussion with the customer.

Going through the requirements and analyzing them has revealed that the project team needs answers to the following questions which were not covered in the requirements document –

- Will the new banking application be used in more than one country?
- Do we have to support multiple languages?
- How many users are expected to use the application?

System Design:

The architect and senior members of the team work on the software architecture, high level, and low-level design for the project.

It is decided that the banking application needs to have redundant backup and failover capabilities such that system is always accessible.

The architect creates the Architecture diagrams and high level / low level design documents.

Implementation:

The development team works on coding the project.

They take the design documents / artifacts and ensure that their solution follows the design finalized by the architect.

Since the application is a banking application and security was a high priority in the application requirements, they implement several security checks, audit logging features in the application.

They also perform several other activities like a senior developer reviewing the other developers code for any issues. Some developers perform static analysis of the code.

Testing:

The testing team tests the complete application and identifies any defects in the application.

These defects are fixed by the developers and the testing team tests the fixes to ensure that the defect is fixed.

They also perform regression testing of the application to see if any new defects were introduced.

Testers with banking domain knowledge were also hired for the project so that they could test the application based on the domain perspective.

Security testing teams were assigned to test the security of the banking application.

Deployment:

The team builds and installs the application on the servers which were procured for the banking application.

Some of the high-level activities include installing the OS on the servers, installing security patches, hardening the servers, installing web servers and application servers, installing the database etc.

They also co-ordinate with network and IT administrative teams to finally get the application up and running on the production servers.

Maintenance:

During the maintenance phase, the team ensures that the application is running smoothly on the servers without any downtime.

Issues that are reported after going live are fixed by the team and tested by the testing team.

4.2 Design Strategy

a. Data Collection: Collect a large and diverse dataset of sign language gestures. This dataset should cover various angles, lighting conditions, and hand shapes. You can collect data through various means such as video recordings, motion capture systems, or real-time capture using cameras.

b. Data Preparation: Once the data is collected, it needs to be preprocessed to make it suitable for training. This involves resizing the images, normalizing the pixel values, and labeling the images with bounding boxes that identify the hand and its associated sign gesture.

c. Training: The next step is to train the YOLOv5 model using the annotated data. The model should be fine-tuned to recognize the hand and the sign language gestures accurately. The model can be trained using transfer learning, where a pre-trained model is used as a starting point and then fine-tuned on the sign language dataset.

d. Testing and Validation: After the model is trained, it needs to be tested and validated on a separate dataset. This dataset should be different from the training dataset to ensure that the model is robust and can generalize well to unseen data.

e. Deployment: Once the model is trained and tested, it can be deployed in a real-world scenario. This involves integrating the model with a camera or a device that can capture the sign language gestures and then recognize them in real-time.

f. Continuous Improvement: Finally, it's essential to continuously improve the model's performance by collecting more data, retraining the model with new data, and fine-tuning its parameters to achieve higher accuracy.

4.3 Module Diagram

UML provides a variety of constructs to represent different kinds of modules. UML has a class construct, which is the object-oriented specialization of a module. Packages can be used in cases where grouping of functionality is important, such as to represent layers and classes.

The purpose of a module diagram is to show the relationship between different components in a system. For UML 2.0, the term " module " refers to a module of classes that represent independent systems or subsystems with the ability to interface with the rest of the system.

There exists a whole development approach that revolves around module: module - based development (CBD). In this approach, module diagrams allow the planner to identify the different module so the whole system does what it's supposed to do.

More commonly, in an OO programming approach, the module diagram allows a senior developer to group classes together based on common purpose so that the developer and others can look at a software development project at a high level.

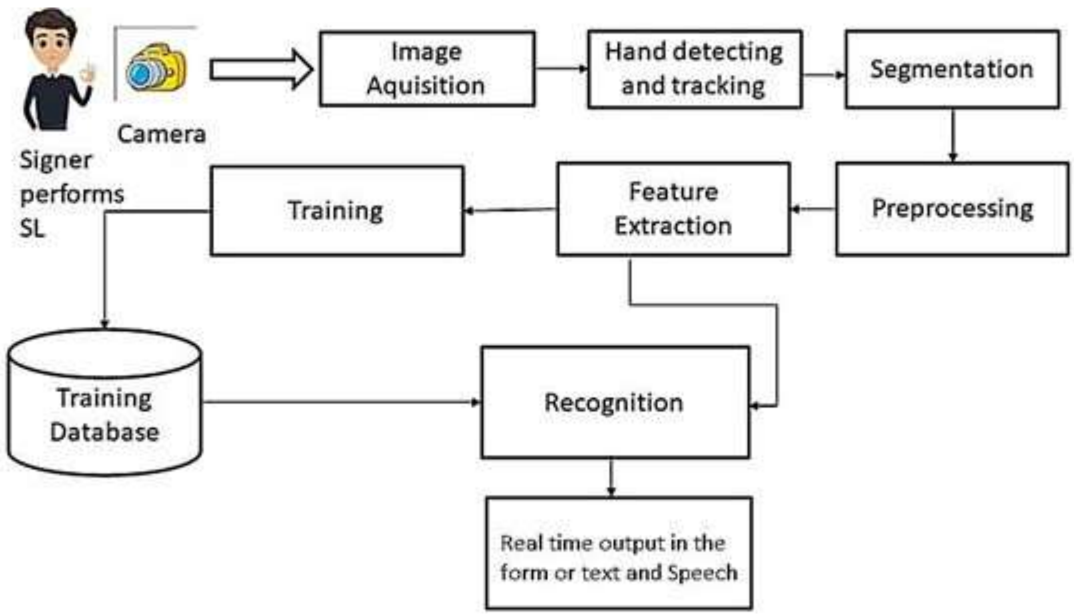


Figure 4.3.1 Module Diagram

4.4 Architecture Diagram

Software Architecture typically refers to biggest structures of a software system, and it deals with how multiple software processes cooperate to carry out their tasks. Software Architecture is described as the organization of a system, where the system represents a set of components that accomplish the defined functions.

A diagram is similar to picture. The architecture diagram examples serve various functions. It always helps the relevant users to learn about system architecture and apply it in the decision-making procedures. It is crucial to communicate information regarding architecture. However, people must follow specific steps before making a diagram for architecture. These are:

- Breaking down communication barriers
- Reaching a consensus
- Decreasing ambiguity

Following figure shows architecture diagram of Application:

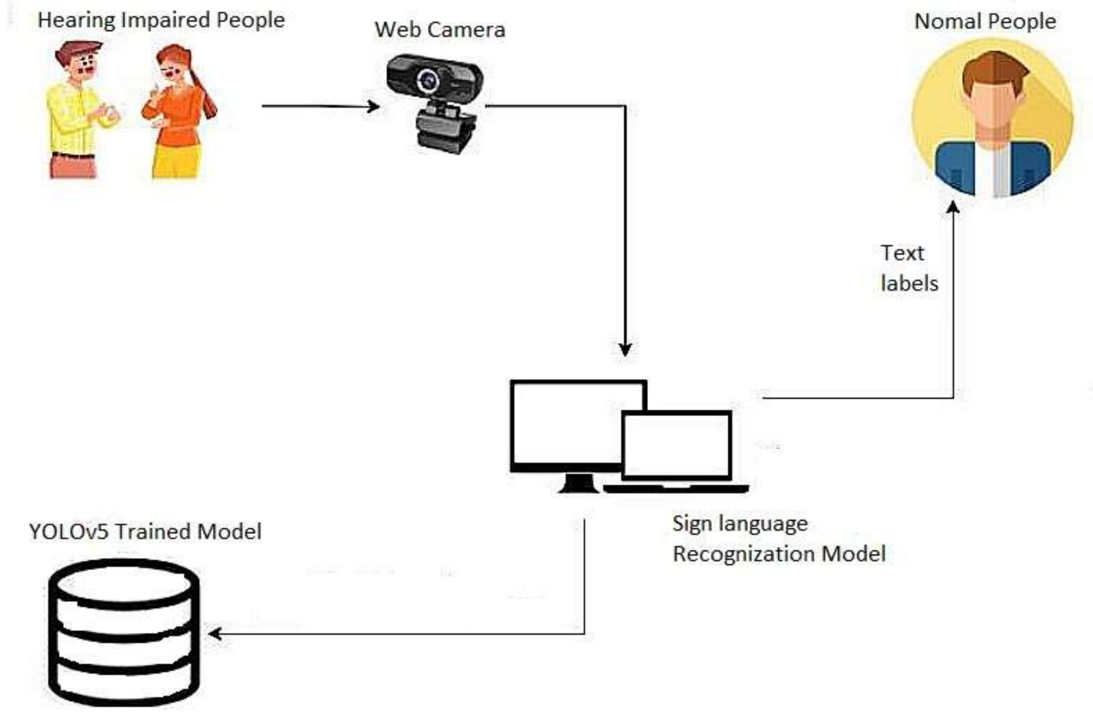


Figure 4.4.1: System Architecture Diagram

An application architecture diagram provides a high-level graphical view of the application architecture, and helps you identify applications, sub-applications, components, databases, services, etc., and their interactions.

A technical architecture diagram provides a bird's eye view of the infrastructure of your organization. The diagram illustrates how components in a system interact with one another in the large scale of things.

The application architecture diagram primarily addresses the "What" in relation to the system.

A common usage of this diagram is to facilitate planning and solutions implementation in the form of assessing the impact of upgrading, replacing or merging existing applications. With new applications continually being released into the market and promising increased efficiency and reduced cost (especially in the containerization space), it's vital to have an overview of the applications within your system.

4.5 Class Diagram

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for Visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

The structure of a system is defined by a Class Diagram by showing its attributes, relationships among objects, and so on. It is the backbone of object-oriented modeling and could also be used for Data modeling. Class Diagrams help in making pre-plans which eases the programming process. Moreover, you could always make changes to the Class Diagram as it's kind of annoying to code different functionality after facts.

It is a design plan based on which a system is built. It is easy to understand without much technical knowledge required.
Class Diagram provides a static view of the application, and its mapping ability with object-oriented language makes it ready to be used in construction. Unlike the sequence diagram, activity diagram, etc., the class diagram is the most popular UML diagram.

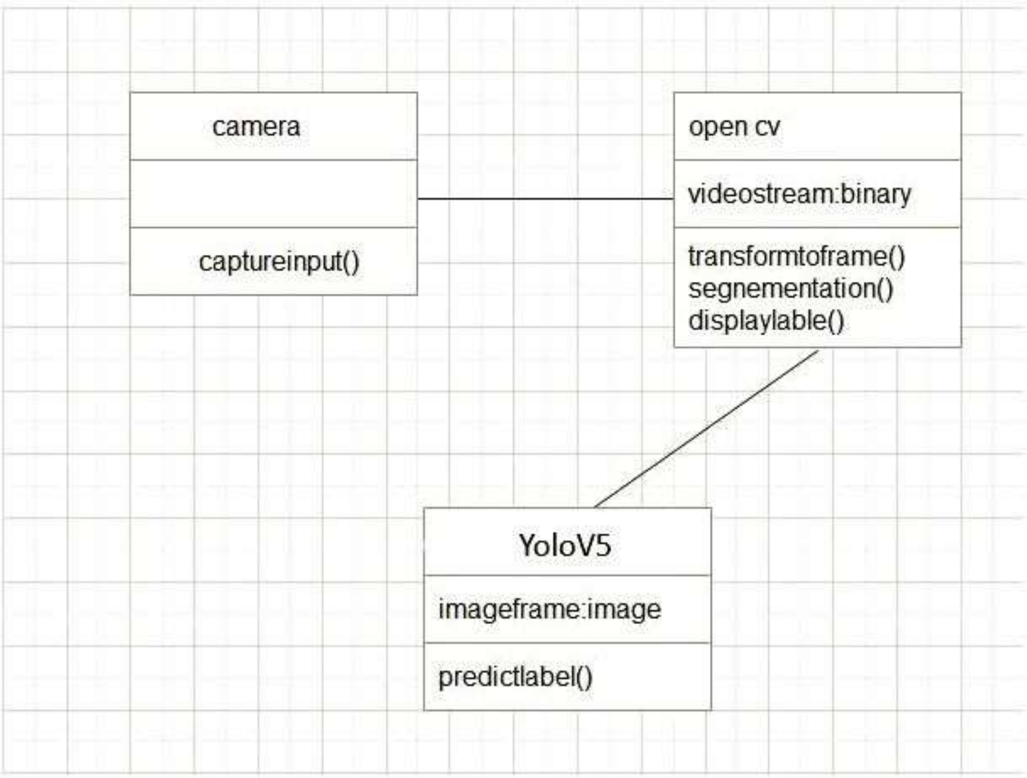


Figure 4.5.1: Class Diagram of Real-Time Assistive system for Deaf-Dumb Community

The class diagram is also considered as the foundation for component and deployment diagrams. Class diagrams are not only used to visualize the static view of the system but are also used to construct the executable code for forward and reverse engineering of any system.

The class diagram clearly shows the mapping with object-oriented languages such as Java, C++, Python etc. From practical experience, a class diagram is generally used for construction purpose.

In a nutshell, it can be said, class diagrams are used for −

- Describing the static view of the system.
- Showing the collaboration among the elements of the static view.
- Describing the functionalities performed by the system.
- Construction of software applications using object-oriented languages.

4.6 Sequence Diagram

UML Sequence diagrams are used to show how objects interact in a given situation an important Characteristics of a sequence diagram is that time passes from top to bottom: the interaction starts near the top of the diagram and ends at the bottom (i.e., Lower equals later.) A popular use for them is to document the dynamics in an -object-oriented system for each Key Collaboration diagrams are created that show how objects interact in various representative scenarios for that collaboration.

Sequence Diagrams captures:

- the interaction that takes place in a collaboration that either realizes a use case or an operation (instance diagrams or generic diagrams)
- high-level interactions between user of the system and the system, between the system and other systems, or between subsystems (sometimes known as system sequence diagrams)

The sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur. Much like the class diagram, developers typically think sequence diagrams were meant exclusively for them. However, an organization's business staff can find sequence diagrams useful to communicate how the business currently works by showing how various business objects interact. Besides documenting an organization's current affairs, a business-level sequence diagram can be used as a requirements document to communicate requirements for a future system implementation. During the requirements phase of a project, analysts can take use cases to the next level by providing a more formal level of refinement. When that occurs, use cases are often refined into one or more sequence diagrams.

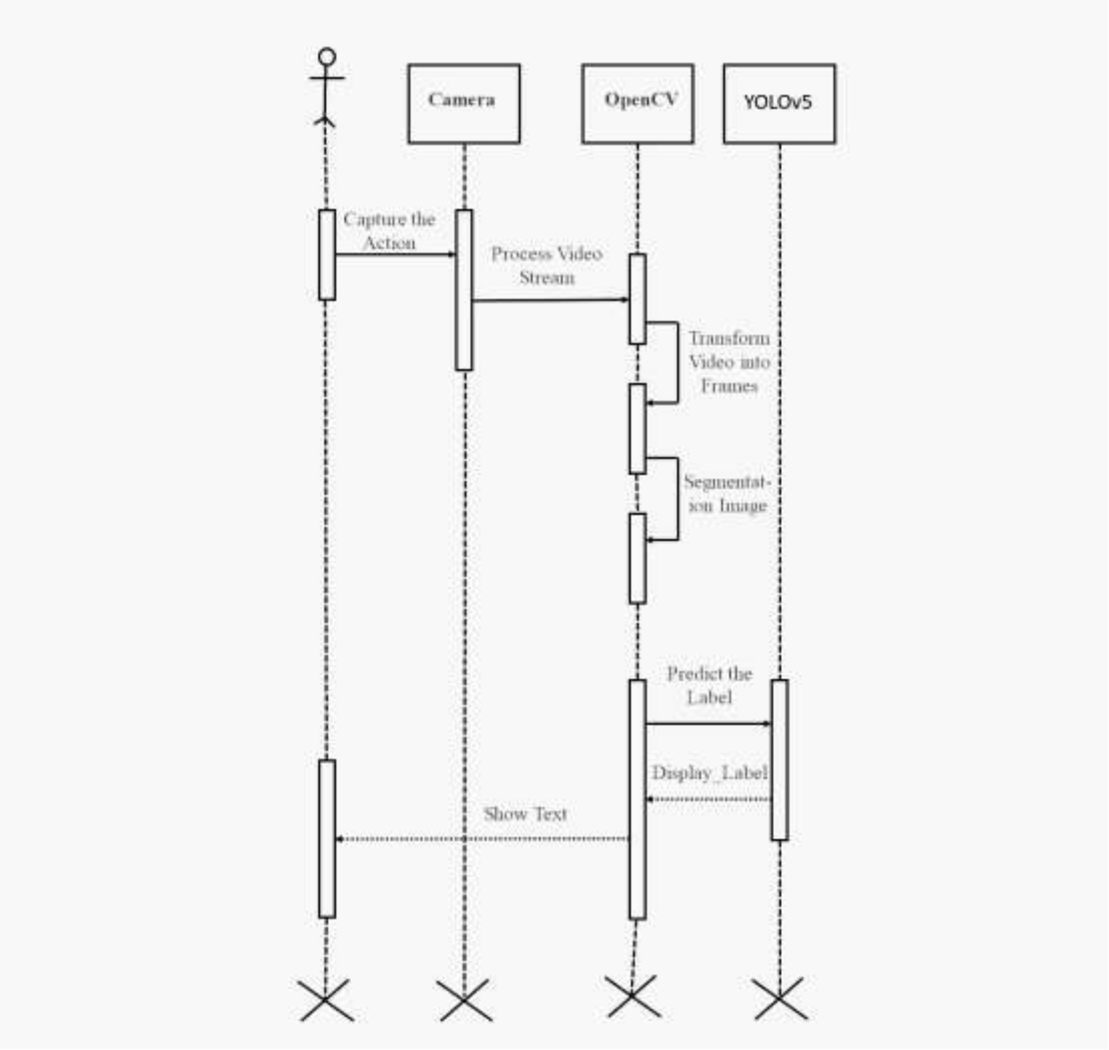


Figure 4.6.1: Sequence Diagram of Real-Time Assistive system for Deaf-Dumb Community

An organization's technical staff can find sequence diagrams useful in documenting how a future system should behave. During the design phase, architects and developers can use the diagram to force out the system's object interactions, thus fleshing out overall system design.

One of the primary uses of sequence diagrams is in the transition from requirements expressed as use cases to the next and more formal level of refinement. Use cases are often refined into one or more sequence diagrams. In addition to their use in designing new systems, sequence diagrams can be used to document how objects in an existing (call it "legacy") system currently interact. This documentation is very useful when transitioning a system to another person or organization.

The main purpose of a sequence diagram is to define event sequences that result in some desired outcome. The focus is less on messages themselves and more on the order in which messages occur; nevertheless, most sequence diagrams will communicate what messages are sent between a system's objects as well as the order in which they occur. The diagram conveys this information along the horizontal and vertical dimensions: the vertical dimension shows, top down, the time sequence of messages/calls as they occur, and the horizontal dimension shows, left to right, the object instances that the messages are sent to.

When doing sequence diagrams, developers love to reuse existing sequence diagrams in their diagram's sequences. Starting in UML 2, the "Interaction Occurrence" element was introduced. The addition of interaction occurrences is arguably the most important innovation in UML 2 interactions modeling. Interaction occurrences add the ability to compose primitive sequence diagrams into complex sequence diagrams. With these you can combine (reuse) the simpler sequences to produce more complex sequences. This means that you can abstract out a complete, and possibly complex, sequence as a single conceptual unit.

4.7 Collaboration Diagram

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.

Collaboration diagrams are created by first identifying the structural elements required to carry out the functionality of an interaction. A model is then built using the relationships between those elements. Several vendors offer software for creating and editing collaboration diagrams.

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features.

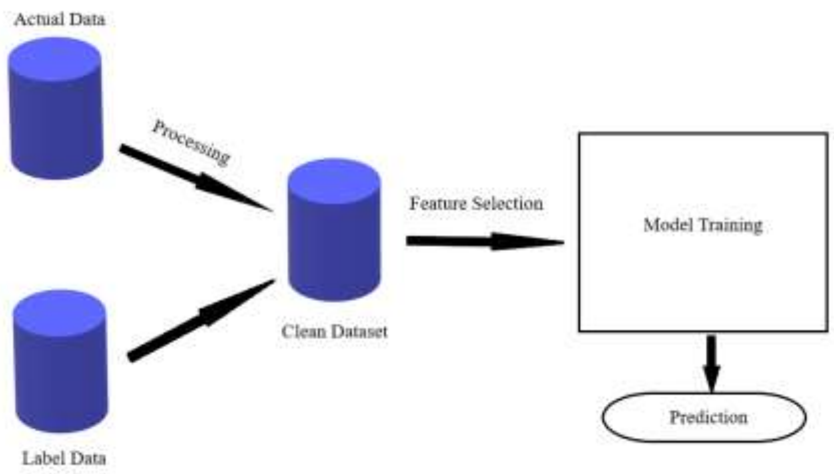


Figure 4.7.1: Collaboration Diagram of Real-Time Assistive system for Deaf-Dumb Community.

Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

The most important objects are placed in the center of the diagram, with all other participating objects branching off. After all objects are placed, links and messages should be added in between. Collaboration diagrams should be used when the relationships among objects are crucial to display.

Collaboration diagrams are like sequence diagrams because of interaction and behavior factors. They are more concerned about object organization rather than sequence diagram that are more focused on a time sequence. They are also known as "Communication Diagram." These are used to represent the flow of messages between the objects.

- The collaboration diagram and sequence diagram show similar information but in a distinct form.
- It can portray the architecture of an object inside the system.
- It can be used to depict the relationship among various objects within the system.
- The collaboration diagram shows the nature of a specific use case.

- They are used to determine the interfaces and class responsibilities.
- Objects collaborate through passing messages to each other.

Uses of the Collaboration Diagram:

- These diagrams design the collaboration among roles and objects that render the functionalities of operations and use cases.
- It models the mechanisms inside an architectural structure of a system.
- The Collaboration diagram models the interactions that represent the passing of messages between the roles and objects inside the collaboration.
- It captures the scenarios in the operations and use cases that contain the collaboration of distinct interactions and objects.
- These diagrams portray the object identification that cooperates in the use cases.
- Every message inside a collaboration diagram contains a sequence number.
- A first message has a sequence number 1. Messages transferred during the call have a similar decimal prefix but have suffixes of 1, 2 accordingly.

4.8 State Chart Diagram

State chart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. State chart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

State chart diagram is used to describe the states of different objects in its life cycle. Emphasis is placed on the state changes upon some internal or external events. These states of objects are important to analyze and implement them accurately.

State chart diagrams are very important for describing the states. States can be identified as the condition of objects when a particular event occurs. Each state diagram typically begins with a dark circle that indicates the initial state and ends with a bordered circle that denotes the final state. However, despite having clear start and end points, state diagrams are not necessarily the best tool for capturing an overall progression of events.

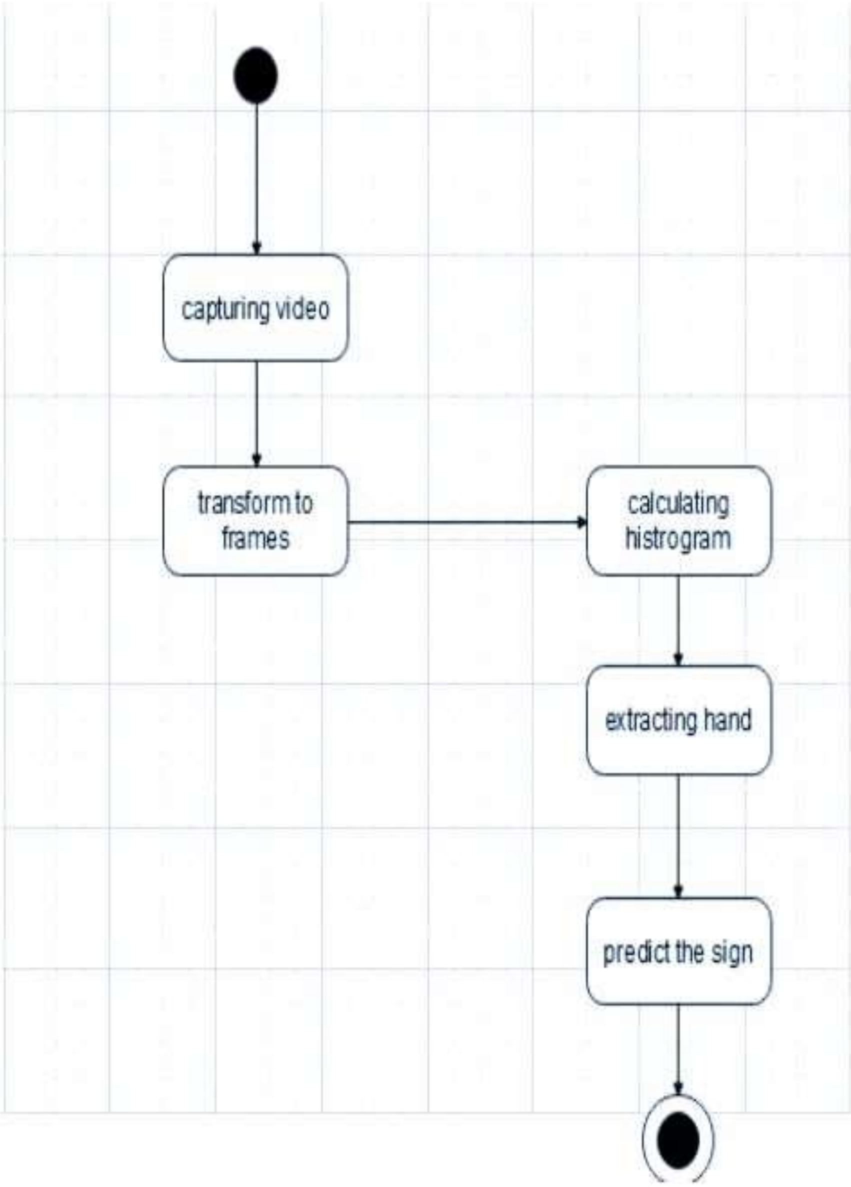


Figure 4.8.1: State chart Diagram of Real-Time Assistive system for Deaf-Dumb Community

Rather, they illustrate specific kinds of behavior—in particular, shifts from one state to another.

State diagrams mainly depict states and transitions. States are represented with rectangles with rounded corners that are labeled with the name of the state. Transitions are marked with arrows that flow from one state to another, showing how the states change.

State diagram applications:

Like most UML diagrams, state diagrams have several uses. The main applications are as follows:

- Depicting event-driven objects in a reactive system.
- Illustrating use case scenarios in a business context.
- Describing how an object moves through various states within its lifetime.
- Showing the overall behavior of a state machine or the behavior of a related set of state machines.

4.9 Activity Diagram

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. So, the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of flow control by using different elements like fork, join etc.

Uses of an Activity Diagram:

- Dynamic modelling of the system or a process.
- Illustrate the various steps involved in a UML use case.
- Model software elements like methods, operations and functions.
- We can use Activity diagrams to depict concurrent activities easily.

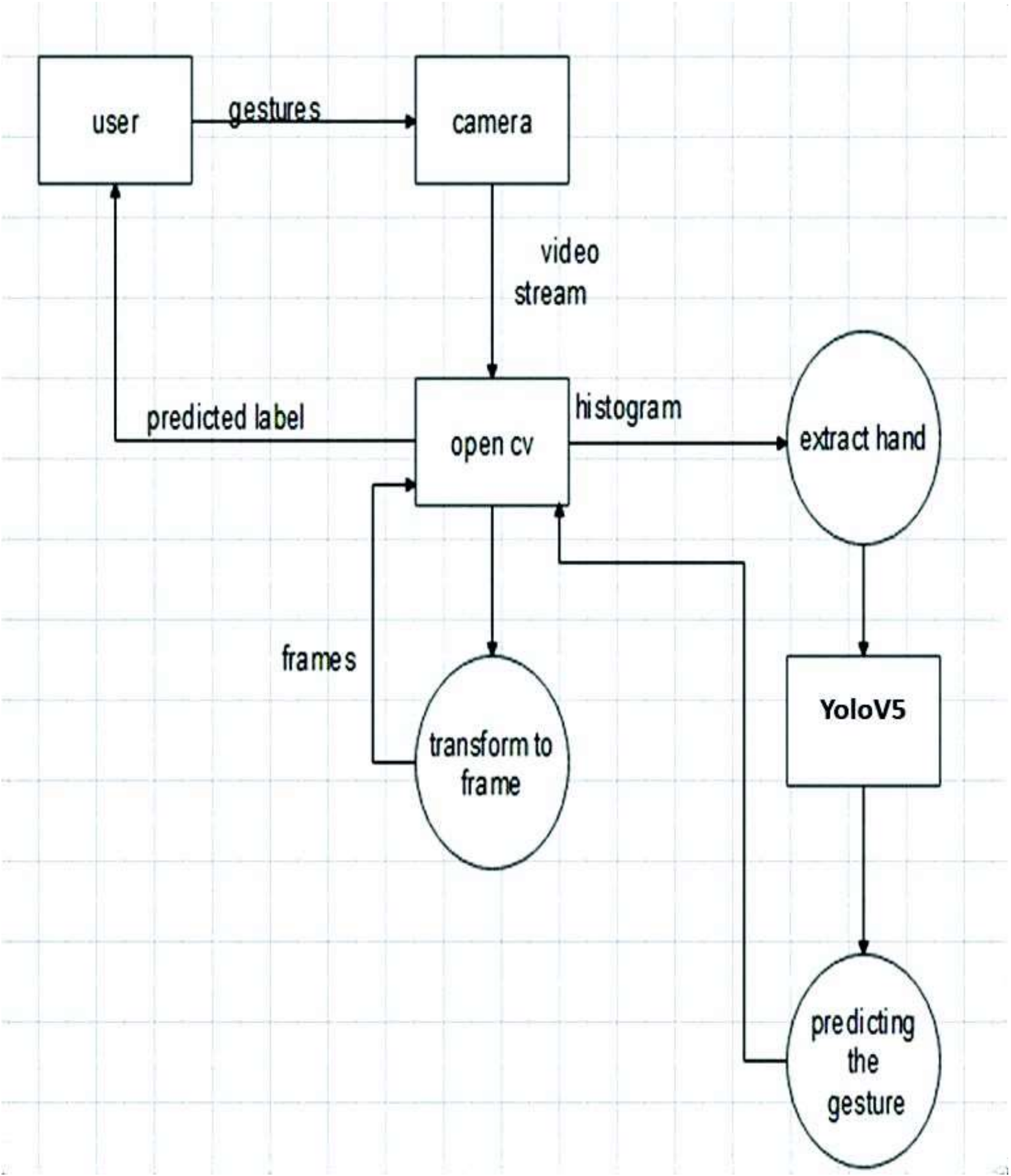


Figure 4.9.1: Activity Diagram of Real-Time Assistive system for Deaf-Dumb Community

The activity diagram helps in envisioning the workflow from one activity to another. It put emphasis on the condition of flow and the order in which it occurs. The flow can be sequential, branched, or concurrent, and to deal with such kinds of flows, the activity diagram has come up with a fork, join, etc.

It is also termed as an object-oriented flowchart. It encompasses activities composed of a set of actions or operations that are applied to model the behavioral diagram.

An activity diagram is used by developers to understand the flow of programs on a high level. It also enables them to figure out constraints and conditions that cause events. A flow chart converges into being an activity diagram if complex decisions are being made. Brevity is the soul of wit. We need to convey a lot of information with clarity and make sure it is short. So, an activity diagram helps people on both sides i.e. Businessmen and Developers to interact and understand systems. A question arises: Do we need to use both the diagram and the textual documentation? Different individuals have different preferences in which they understand something. For example: To understand a concept, some people might prefer a written tutorial with images while others would prefer a video lecture. So, we generally use both the diagram and the textual documentation to make our system description as clear as possible. We also need to be sensitive to the needs of the audience that we are catering to at times. Difference between a Use case diagram and an activity diagram.

An activity diagram is used to model the workflow depicting conditions, constraints, sequential and concurrent activities. On the other hand, the purpose of a Use Case is to just depict the functionality i.e., what the system does and not how it is done. So, in simple terms, an activity diagram shows 'How' while a Use case shows 'What' for a particular system. The levels of abstraction also vary for both. An activity diagram can be used to illustrate a business process (high level implementation) to a stand-alone algorithm (ground level implementation). However, use cases have a low level of abstraction. They are used to show a high level of implementation only.

5. IMPLEMENTATION

5.1 Implementation Strategy

Data collection: Collect a dataset of sign language gestures, preferably including a diverse set of signers, backgrounds, and lighting conditions. Annotate the dataset with bounding boxes around each sign gesture.

- **Gathering a representative sample of sign language gestures**: The first step in data collection is to identify the set of sign language gestures that you want to recognize.
- **Capturing images of sign language gestures**: Once you have identified the set of gestures to be recognized, you will need to capture images of people performing those gestures. This can be done using a camera. It is important to capture the images in a well-lit environment and from multiple angles to capture the variability of the gestures.
- **Labeling the Images**: The next step is to label the images with bounding boxes around the hands and fingers of the sign language user. This is typically done manually by drawing the boxes using a tool such as LabelImg. Each bounding box is associated with a class label corresponding to the sign language gesture being performed in that frame.
- **Splitting the data into training and validation sets**: The labeled images are then split into two sets: a training set and a test set. The training dataset is used to train the YOLOv5 model to recognize the sign language gestures, while the test dataset is used to evaluate the performance of the model.
- **Augmenting the data**: To improve the generalization ability of the model, data augmentation techniques such as random cropping, rotation, and flipping may be applied to the training set.
- **Converting the data into YOLOv5 format**: Finally, the labeled and augmented data is converted into the YOLOv5 format, which consists of text files containing the coordinates and class labels of the bounding boxes for each frame in the images. These files are used to train the YOLOv5 model using a

deep learning framework such as PyTorch or TensorFlow.

Data preparation: Preprocess the data by resizing the images to a uniform size and converting them to a format that YOLOv5 can use, such as COCO or YOLO format.

- **Creating a YAML file*:*** A YAML file is used to specify the configuration of the model. It includes the location of the training and validation data, the number of classes, and the anchors used for object detection.
- **Converting the data to YOLOv5 format*:*** The labeled data needs to be converted to the YOLOv5 format using the `create_dataset.py` script. This script takes the labeled data and generates a YOLOv5-compatible dataset.
- **Training the model*:*** Once the dataset is created, the model can be trained using the `train.py` script. The script takes the YAML file and the dataset as input and trains the model using the YOLOv5 algorithm.
- **Model Selection**: Choose a pre-trained YOLOv5 model that has been trained on a similar type of image data, such as COCO or VOC. You can use the pre-trained model as a starting point and fine-tune it on your own dataset.
- **Fine Tuning**: Fine-tune the pre-trained model on your own dataset using transfer learning. This involves updating the weights of the pre-trained model to learn the specific features of your sign language dataset. You can also adjust the hyperparameters of the model to optimize its performance on your specific task.
- **Training**: Train the fine-tuned model on your dataset. This involves feeding the images into the model and using a loss function to measure the difference between the predicted bounding boxes and the ground truth bounding boxes.
- **Evaluation**: Evaluate the performance of the trained model on a validation set to see how well it can generalize to new sign language gestures.
- **Deployment**: Deploy the trained model to a device or platform where it can be used for real-time sign language recognition.
- **Integration**: Integrate the model with a user interface or application that allows users to interact with the system, such as a mobile app or web interface.

5.2 Hardware Platform Used

In this project, a computer with sufficient processing power is needed. This project also requires webcam or inbuild system camera.

- System: Intel Processor i3/i5/i7 or AMD processors
- RAM
- Hard Disk: 1GB
- Webcam

In this project the system needed to collect real time data from soil sample which needs be captured over period of time and submitted to a system for predictions. Hence we used following component:

5.2.1 Webcam:

A webcam, short for web camera, is a type of digital camera that is designed to capture video and transmit it in real-time over the internet. It is typically used for video conferencing, online meetings, live streaming, and other online communication applications.

Webcams are usually small and lightweight, and they can be attached to a computer or laptop via USB or other connection ports. They contain a lens and an image sensor that captures video and sends it to the computer for processing and transmission.

Figure 5.4.1: Webcam

Webcams may also come equipped with a microphone to capture audio, and some models feature additional features such as autofocus, zoom, and pan capabilities.

Webcams are essential tools for remote communication and collaboration, enabling people to connect and communicate with each other from anywhere in the world. They have become increasingly popular in recent years as more and more people work and learn remotely, and they have become an indispensable part of modern communication technology.

Webcams can be used for sign language detection in several ways. One common approach is to use computer vision and machine learning techniques to analyze the video feed from the webcam and recognize hand gestures and signs. This involves training a machine learning model on a dataset of sign language gestures and then using the webcam to capture new video data and classify the hand gestures in real-time.

Another approach is to use depth sensing technology, such as Microsoft's Kinect or

Intel's RealSense cameras, which can capture 3D information about the hand and its movements. This allows for more accurate and robust detection of sign language gestures, as it can capture the depth and position of the hand in space.

Webcams can also be used in conjunction with sign language recognition software to enable real-time translation of sign language into spoken or written text. This can be useful for deaf or hard-of-hearing individuals who communicate primarily through sign language, as it allows them to communicate more easily with hearing individuals who may not know sign language.

Overall, webcams can be a valuable tool for sign language detection and translation, as they provide a convenient and accessible means of capturing video data and analyzing sign language gestures in real-time.

5.3 Software Platform Used

- Python
- NumPy
- OpenCV
- YOLOv5
- TensorFlow
- VS-code

In this project we used various python libraries to create machine learning model, also used machine learning opensource model.

5.3.1 YOLOv5

YOLOv5 (You Only Look Once version 5) is a state-of-the-art deep learning model for object detection developed by Ultralytics. It is an improvement upon the previous versions of YOLO, with several enhancements that make it faster and more accurate. YOLOv5 uses a deep neural network to detect objects in images and video frames. The model divides the image into a grid of cells and predicts bounding boxes and confidence scores for each cell. The predicted bounding boxes indicate the location of an object in the image, while the confidence score indicates how certain the model is that the object is present.

One of the key features of YOLOv5 is its architecture. The model is based on a convolutional neural network that uses a series of convolutional layers to extract features from the input image. The features are then passed through several fully connected layers to predict the bounding boxes and confidence scores. Another key feature of YOLOv5 is its speed. The model is designed to be highly optimized for both CPU and GPU hardware, allowing it to process images and video frames in real-time. This makes it suitable for applications like autonomous vehicles, security cameras, and robotics.

YOLOv5 also has several improvements in accuracy compared to previous versions. The model uses a novel anchor box selection algorithm that allows it to detect small objects more accurately. It also includes several enhancements to the loss function that improve the training process and make the model more robust to different types of images and environments. Overall, YOLOv5 is a highly accurate and efficient deep learning model for object detection. It has several advantages over traditional object detection models, including faster inference times, higher accuracy, and the ability to detect small objects.

5.3.2 NumPy

NumPy (Numerical Python) is a popular open-source library for numerical computing in Python. NumPy provides an array object that is efficient for storing and manipulating large multidimensional arrays and matrices, as well as a variety of functions for performing mathematical operations on these arrays. One of the key advantages of NumPy is that it enables fast mathematical computations on large datasets by leveraging optimized C and Fortran libraries.

This makes it an essential tool for scientific computing and data analysis. In addition to its array object, NumPy provides a wide range of functions for performing mathematical operations such as addition, subtraction, multiplication, division, and trigonometric functions, among others. It also includes tools for linear algebra, Fourier analysis, and random number generation.

NumPy is particularly useful in data analysis and machine learning tasks, where large datasets need to be manipulated and processed efficiently. It is also often used in conjunction with other scientific libraries such as SciPy, Pandas, and Matplotlib. Overall, NumPy is a powerful and widely used library for scientific computing in Python that provides efficient and easy-to-use tools for working with numerical data.

5.3.3 OpenCV

OpenCV (Open-Source Computer Vision Library) is a free, open-source computer vision and machine learning software library designed to help developers create applications that can interpret and understand the visual world. OpenCV was initially developed by Intel in 1999 and is now maintained by the OpenCV community. It has been written in C++, but also has bindings for Python and other programming languages.

OpenCV provides a wide range of tools and functions for image and video processing, including basic image manipulation, feature detection and matching, object recognition and tracking, and machine learning algorithms. It also provides real-time computer vision capabilities, making it suitable for applications such as augmented reality, face recognition, and robotics. OpenCV has a rich documentation and community support, which makes it easy for developers to learn and use the library. It is cross-platform and runs on different operating systems such as Windows, Linux, and macOS. Overall, OpenCV is a powerful and versatile library that has been widely used in various fields such as robotics, automotive, medical imaging, and entertainment.

5.3.4 Pandas

Pandas is a popular open-source Python library for data manipulation, analysis, and preparation. It provides a fast and flexible data structure, called a Data Frame, which allows users to work with labeled and structured data in a natural way. Pandas is built on top of NumPy and provides several functions that make it easy to manipulate and analyze data. Some of the key features of pandas include:

Data ingestion: Pandas can read data from a variety of file formats, including CSV, Excel, SQL databases, and more.

Data cleaning: Pandas provide tools for cleaning and handling missing or corrupted data, such as dropping or filling missing values.

Data manipulation: Pandas provide functions for transforming and reshaping data, such as merging, grouping, pivoting, and sorting.

Data analysis: Pandas provide functions for statistical analysis, such as calculating summary statistics, applying functions to data, and performing time series analysis. Pandas also integrates well with other Python libraries, such as Matplotlib for visualization and Scikit-learn for machine learning. Overall, Pandas is a powerful and flexible library that makes it easy for users to work with structured data and perform data analysis tasks in Python. It is widely used in data science and data analysis projects.

5.3.5 TensorFlow

TensorFlow is an open-source software library developed by the Google Brain team for machine learning and artificial intelligence applications. It was first released in 2015 and has since become one of the most popular deep learning frameworks. TensorFlow provides a flexible and comprehensive platform for building and training various types of machines learning models, including neural networks. It supports a wide range of applications, including image and speech recognition, natural language processing, and robotics.

The library uses a data flow graph model, where nodes in the graph represent mathematical operations, and edges represent the input/output relationships between those operations. This allows TensorFlow to efficiently distribute computation across multiple CPUs or GPUs, making it suitable for large-scale machine learning applications. TensorFlow also provides high-level APIs for building and training neural networks, such as Keras, which simplifies the process of model development and training. It also has a rich ecosystem of tools and libraries, including Tensor Board for visualizing and debugging models, and TensorFlow Lite for deploying models on mobile and embedded devices. Overall, TensorFlow is a powerful and flexible framework that has been widely used in various fields such as healthcare, finance, and robotics, and has contributed to significant advancements in machine learning and artificial intelligence.

5.3.6 Requests

The Requests library is a popular open-source Python library for making HTTP requests to web servers. It simplifies the process of sending HTTP/1.1 requests, handling headers, payloads, authentication, and more. The Requests library provides a simple and intuitive API for performing common HTTP methods such as GET, POST, PUT, DELETE, and PATCH. It supports HTTP and HTTPS protocols and can be used to interact with RESTful web services, web applications, and other web-based resources.

The library also allows for customization of HTTP headers and cookies, handling of sessions, and authentication methods such as basic, digest, and OAuth. It can also handle redirects and timeouts, making it suitable for handling various network scenarios. Requests is built on top of urllib3, which provides connection pooling and thread safety features, allowing for efficient use of network resources.

Overall, the Requests library is a convenient and efficient tool for making HTTP requests in Python, and its simple syntax and versatility make it a popular choice for developers working with web-based APIs and services.

5.3.7 JSON

import json is a Python statement that imports the built-in json module. The json module is a part of the Python standard library and provides support for encoding and decoding JSON (JavaScript Object Notation) formatted data. JSON is a lightweight data interchange format that is easy for humans to read and write, and easy for machines to parse and generate. It is widely used for transmitting and storing data between web applications and APIs. The json module in Python provides two main methods for working with JSON data: json. loads(): This method is used to decode a JSON-formatted string into a Python object. For example, json. loads('{"name": "John", "age": 30}') would return a Python dictionary object with the keys "name" and "age".

json.dumps(): This method is used to encode a Python object into a JSON-formatted string. For example, json.dumps({"name": "John", "age": 30}) would return a JSON-formatted string {"name": "John", "age": 30}. The json module also provides several other functions and classes for working with JSON data, such as json.load() and json.dump() for working with JSON files, and json.JSONEncoder and json.JSONDecoder for customizing JSON encoding and decoding.

Overall, the json module in Python provides a simple and easy-to-use interface for encoding and decoding JSON data, making it a popular choice for working with web APIs and data interchange formats.

5.3.8 YAML

import yaml is a Python statement that imports the PyYAML library. PyYAML is a popular Python library for working with YAML (YAML Ain't Markup Language) formatted data. YAML is a human-readable data serialization format that is often used for configuration files, data exchange between languages, and data storage. The yaml module in Python provides two main methods for working with YAML data:

yaml.load(): This method is used to load YAML-formatted data from a string or a file and convert it into a Python object. For example, yaml.load('name: John\nage: 30\n') would return a Python dictionary object with the keys "name" and "age".

yaml.dump(): This method is used to convert a Python object into YAML-formatted data and write it to a file or a string. For example, yaml.dump({"name": "John", "age": 30}, file) would write the YAML-formatted data name: John\nage: 30\n to a file.

The yaml module also provides several other functions and classes for working with YAML data, such as yaml.load_all() and yaml.dump_all() for working with multiple YAML documents, and yaml.SafeLoader and yaml.SafeDumper for safe loading and dumping of YAML data. Overall, the yaml module in Python provides a flexible and easy-to-use interface for working with YAML data, making it a popular choice for configuration files and data serialization in Python projects.

5.3.9 Platform

import platform is a Python statement that imports the built-in platform module. The platform module is a part of the Python standard library and provides a way to access various system-specific parameters and functions, such as the operating system, hardware architecture, and Python version. Some of the common tasks that can be performed using the platform module include:

Getting the operating system name and version: The platform.system() and platform.release() functions can be used to get the name and version of the operating system running the Python interpreter.

Getting the Python version: The platform.python_version() function can be used to get the version of the Python interpreter.

Getting the machine hardware architecture: The platform.machine() and platform.architecture() functions can be used to get information about the hardware architecture of the machine running the Python interpreter.

Getting the network node name: The platform.node() function can be used to get the network node name of the machine running the Python interpreter.

The platform module also provides several other functions and classes for working with system-specific information, such as platform.uname() for getting information about the underlying hardware platform, and platform.processor() for getting information about the processor of the machine running the Python interpreter.

Overall, the platform module in Python provides a convenient way to access system-specific information and can be useful for writing platform-independent Python code that adapts to different operating systems and hardware architectures.

5.3.10 Sys Library

import sys is a Python statement that imports the built-in sys module. The sys module is a part of the Python standard library and provides access to some system-specific parameters and functions.Some of the common tasks that can be performed using the sys module include:Accessing command-line arguments: The sys.argv attribute provides access to the command-line arguments passed to a Python script. Exiting the program: The sys.exit() function can be used to exit a Python program with a given exit status.Accessing system-specific information: The sys.platform attribute provides information about the operating system running the Python interpreter. Controlling the Python interpreter: The sys.stdin, sys.stdout, and sys.stderr attributes provide access to the standard input, standard output, and standard error streams, respectively.The sys module is also used by other built-in Python modules and third-party libraries, and it can be useful for debugging and troubleshooting Python scripts.

Overall, the sys module is a useful tool for accessing system-specific information and controlling the behavior of the Python interpreter.

5.3.11 OS

import os is a Python statement that imports the built-in os module. The os module provides a way to interact with the operating system in a platform-independent way, allowing Python code to access and manipulate files and directories, as well as execute system commands.Some of the common tasks that can be performed using the os module include: Working with files and directories: The os module provides functions for creating, deleting, moving, and renaming files and directories, as well as for checking file and directory properties such as size, modification time, and permissions. Some of the commonly used functions for working with files and directories include os.path.join(), os.path.exists(), os.listdir(), os.mkdir(), os.rmdir(), and os.rename().

Running system commands: The os module provides functions for running system commands, such as os.system(), os.popen(), and os.spawn(). These functions can be used to execute system commands and capture their output or exit status.Managing environment variables: The os module provides functions for getting and setting environment variables, such as os.environ.get() and os.environ.update().

Working with processes: The os module provides functions for managing processes, such as os.fork(), os.exec(), and os.kill(). These functions can be used to create and manage child processes. The os module also provides several other functions and constants for working with the operating system, such as os.pathsep for getting the path separator, os.name for getting the operating system name, and os.getcwd() for getting the current working directory. Overall, the os module in Python provides a powerful and flexible way to interact with the operating system, making it a fundamental part of many Python programs that deal with file and directory operations, system administration tasks.

5.4 Deployment Diagram

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system. Using it you can understand how the system will be physically deployed on the hardware. Deployment diagrams help model the hardware topology of a system compared to other UML diagram types which mostly outline the logical components of a system.

Deployment diagram represents the deployment view of a system. It is related to the component diagram because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware used to deploy the application.

Deployment diagrams are useful for system engineers. An efficient deployment diagram is very important as it controls the following parameters −

- Performance
- Scalability
- Maintainability
- Portability

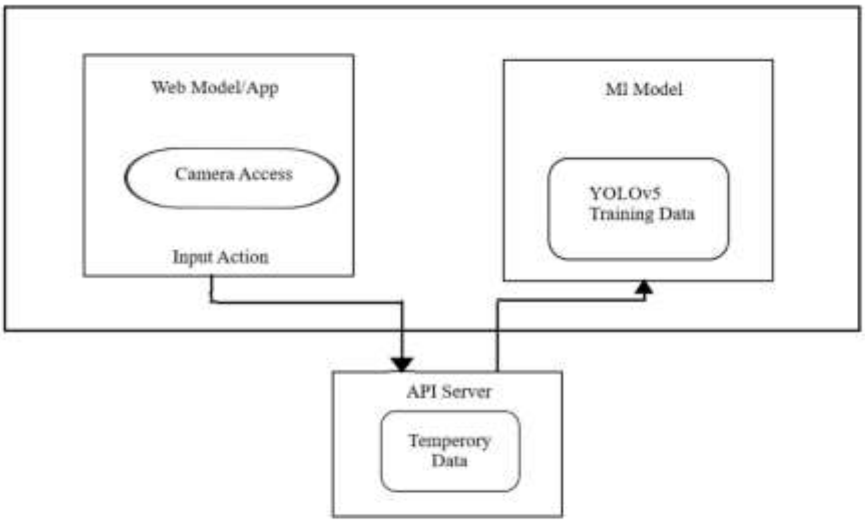


Figure 5.6.1: Deployment Diagram of Application

5.5 Implementation Level Details

The implementation of a sign language detection system using YOLOv5 involves several algorithms for various tasks such as image pre-processing, object detection, and gesture recognition. Here are some of the key algorithms used in the system:

- Image Pre-processing:

Before feeding the images into the object detection algorithm, it is essential to preprocess the images. Some of the common preprocessing steps include:
- Resizing the image to a standard size to reduce the computational cost.
- Normalizing the pixel values to be between 0 and 1 to improve the convergence rate of the model.
- Applying data augmentation techniques like flipping, rotating, and scaling to generate more training data and improve model performance.

- Object Detection using YOLOv5:

The core algorithm used in the sign language detection system is the YOLOv5 algorithm for object detection. YOLOv5 is a state-of-the-art object detection algorithm that uses a deep neural network to detect objects in real-time. The algorithm consists of the following key components:

- Backbone Network: A deep convolutional neural network that extracts features from the input image.
- Neck Network: A network that combines the features extracted by the backbone network to generate a high-level feature map.
- Head Network: A network that predicts the bounding boxes and class labels for each object in the input image.

- Gesture Recognition:

Once the sign language gestures are detected, they need to be recognized to interpret the sign language message. Gesture recognition involves the following steps:

- Segmentation: The detected hand region is segmented from the image to isolate the hand and improve gesture recognition accuracy.
- Feature Extraction: Various features of the hand, such as the shape, size, and orientation, are extracted to represent the sign language gesture.
- Classification: A machine learning algorithm, such as a support vector machine (SVM), is used to classify the gesture based on the extracted features.

- Post-processing:

After the gesture is recognized, it needs to be combined with other detected signs and processed to generate the final sign language message. This may involve additional processing such as smoothing and filtering of the detected gestures.

Overall, the sign language detection system using YOLOv5 involves several algorithms, including image pre-processing, object detection, gesture recognition, and post-processing. These algorithms work together to accurately detect sign language gestures and interpret them into meaningful messages.

5.6 Testing

The purpose of testing is to discover errors. Testing is a process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. Software testing is an important element of the software quality assurance and represents the ultimate review of specification, design and coding. The increasing feasibility of software as a system and the cost associated with the software failures are motivated forces for well planned through testing.

Testing Objectives*:* There are several rules that can serve as testing objectives they are:

- Testing is a process of executing program with the intent of finding an error.
- A good test case is the one that has a high probability of finding an undiscovered error.

Types of Testing:

To make sure that the system does not have errors, the different levels of testing strategies that are applied at different phases of software development are:

Unit Testing

Unit testing is done at the lowest level. It tests the basic unit of software, that is the smallest testable piece of software. The individual component or unit of a program are tested in unit testing.

Unit Testing is a software testing technique by means of which individual units of software i.e., group of computer program modules, usage procedures and operating procedures are tested to determine whether they are suitable for use or not. It is a testing method using which every independent modules are tested to determine if there are any issue by the developer himself. It is correlated with functional correctness of the independent modules.

Unit Testing is defined as a type of software testing where individual components of a software are tested.

Unit Testing of software product is carried out during the development of an application. An individual component may be either an individual function or a procedure. Unit Testing is typically performed by the developer.

In SDLC or V Model, Unit testing is first level of testing done before integration testing. Unit testing is such type of testing technique that is usually performed by the developers. Although due to reluctance of developers to tests, quality assurance engineers also do unit testing.

Objective of Unit Testing:

The objective of Unit Testing is:

- To isolate a section of code.
- To verify the correctness of code.
- To test every function and procedure.
- To fix bug early in development cycle and to save costs.
- To help the developers to understand the code base and enable them to make changes quickly.
- To help for code reuse.

Unit Testing Tools:

Here are some commonly used Unit Testing tools:

- Jtest
- Junit
- NUnit
- EMMA
- PHPUnit

Advantages of Unit Testing:

- Unit Testing allows developers to learn what functionality is provided by a unit and how to use it to gain a basic understanding of the unit API.
- Unit testing allows the programmer to refine code and make sure the module works properly.
- Unit testing enables to test parts of the project without waiting for others to be completed.

Unit testing are of two types.

- Black box testing: This is also known as functional testing, where the test cases are designed based on input output values only. The black-box approach is a testing method in which test data are derived from the specified functional requirements without regard to the final program structure. It is also termed data-driven, input/output driven, or requirements-based testing. Because only the functionality of the software module is of concern, black-box testing also mainly refers to functional testing -- a testing method emphasized on executing the functions and examination of their input and output data. The tester treats the software under test as a black box -- only the inputs, outputs and specification are visible, and the functionality is determined by observing the outputs to corresponding inputs. In testing, various inputs are exercised and the outputs are compared against specification to validate the correctness. All test cases are derived from the specification. No implementation details of the code are considered.

It is obvious that the more we have covered in the input space, the more problems we will find and therefore we will be more confident about the quality of the software. Ideally, we would be tempted to exhaustively test the input space. But as stated above, exhaustively testing the combinations of valid inputs will be impossible for most of the programs, let alone considering invalid inputs, timing, sequence, and resource variables. Combinatorial explosion is the major roadblock in functional testing. To make things worse, we can never be sure whether the specification is either correct or complete. Due to limitations of the language used in the specifications (usually natural language), ambiguity is often inevitable. Even if we use some type of formal or restricted language, we may still fail to write down all the possible cases in the specification. Sometimes, the specification itself becomes an intractable problem: it is not possible to precisely specify every situation that can be encountered using limited words. And people can seldom specify clearly what they want -- they usually can tell whether a prototype is, or is not, what they want after they have been finished. Specification problems contributes approximately 30 percent of all bugs in software.

The research in black box testing mainly focuses on how to maximize the effectiveness of testing with minimum cost, usually the number of test cases. It is not possible to exhaust the input space, but it is possible to exhaustively test a subset of the input space. Partitioning is one of the common techniques. If we have partitioned the input space and assume all the input values in a partition is equivalent, then we only need to test one representative value in each partition to sufficiently cover the whole input space. Domain testing partitions the input domain into regions, and consider the input values in each domain an equivalent class. Domains can be exhaustively tested and covered by selecting a representative value(s) in each domain. Boundary values are of special interest. Experience shows that test cases that explore boundary conditions have a higher payoff than test cases that do not. Boundary value analysis requires one or more boundary values selected as representative test cases. The difficulties with domain testing are that incorrect domain definitions in the specification cannot be efficiently discovered.

Good partitioning requires knowledge of the software structure. A good testing plan will not only contain black-box testing, but also white-box approaches, and combinations of the two.

There are many types of Black Box Testing but following are the prominent ones.

- Equivalence class partitioning: In this approach, the domain of input values to a program is divided into a set of equivalence classes. e.g., Consider a software program that computes whether an integer number is even or not that is in the range of 0 to 10. Determine the equivalence class test suite. There are three equivalence classes for this program. - The set of negative integers - The integers in the range 0 to 10 - The integer larger than 10.

- Boundary value analysis: In this approach, while designing the test cases, the values at boundaries of different equivalence classes are taken into consideration. e.g. In the above given example as in equivalence class partitioning, a boundary values based test suite is {0, -1, 10, 11}

- White box testing: It is also known as structural testing. In this testing, test cases are designed based on examination of the code. This testing is performed based on the knowledge of how the system is implemented. It includes analyzing data flow, control flow, information flow, coding practices, exception, and error handling within the system, to test the intended and unintended software behavior. White box testing can be performed to validate whether code implementation follows intended design, to validate implemented security functionality, and to uncover exploitable vulnerabilities. This testing requires access to the source code. Though white box testing can be performed any time in the life cycle after the code is developed, but it is a good practice to perform white box testing during the unit testing phase.

Contrary to black-box testing, software is viewed as a white-box, or glass-box in white-box testing, as the structure and flow of the software under test are visible to the tester. Testing plans are made according to the details of the software implementation, such as programming language, logic, and styles. Test cases are derived from the program structure. White-box testing is also called glass-box testing, logic-driven testing or design-based testing.

There are many techniques available in white-box testing, because the problem of intractability is eased by specific knowledge and attention on the structure of the software under test. The intention of exhausting some aspect of the software is still strong in white-box testing, and some degree of exhaustion can be achieved, such as executing each line of code at least once (statement coverage), traverse every branch statement (branch coverage), or cover all the possible combinations of true and false condition predicates (Multiple condition coverage).

Control-flow testing, loop testing, and data-flow testing, all maps the corresponding flow structure of the software into a directed graph. Test cases are carefully selected based on the criterion that all the nodes or paths are covered or traversed at least once. By doing so we may discover unnecessary "dead" code -- code that is of no use, or never get executed at all, which cannot be discovered by functional testing.

In mutation testing, the original program code is perturbed and many mutated programs are created, each contains one fault. Each faulty version of the program is called a mutant. Test data are selected based on the effectiveness of failing the mutants. The more mutants a test case can kill, the better the test case is considered. The problem with mutation testing is that it is too computationally expensive to use. The boundary between black-box approach and white-box approach is not clear-cut. Many testing strategies mentioned above, may not be safely classified into black-box testing or white-box testing. It is also true for transaction-flow testing, syntax testing, finite-state testing, and many other testing strategies not discussed in this text. One reason is that all the above techniques will need some knowledge of the specification of the software under test. Another reason is that the idea of specification itself is broad -- it may contain any requirement including the structure, programming language, and programming style as part of the specification content.

We may be reluctant to consider random testing as a testing technique. The test case selection is simple and straightforward: they are randomly chosen. Study in indicates that random testing is more cost effective for many programs. Some very subtle errors can be discovered with low cost. And it is also not inferior in coverage than other carefully designed testing techniques. One can also obtain reliability estimate using random testing results based on operational profiles. Effectively combining random testing with other testing techniques may yield more powerful and cost-effective testing strategies.

Integration Testing

Integration testing is performed when two or more tested units are combined into a larger structure. The main objective of this testing is to check whether the different modules of a program interface with each other properly or not. It is is the process of testing the connectivity or data transfer between a couple of unit tested modules. It is AKA I&T Testing or String Testing. It is subdivided into the Top-Down Approach, Bottom-Up Approach, and Sandwich Approach (Combination of Top-Down and Bottom-Up). This testing is mainly of two types:

- Top-down approach
- Bottom-up approach

In bottom-up approach, each subsystem is tested separately and then the full system is tested. But the top-down integration testing starts with the main routine and one or two subordinate routines in the system. After the top-level 'skeleton' has been tested, the immediately subroutines of the 'skeleton' are combined with it and tested.

1. **Big-Bang Integration Testing –** It is the simplest integration testing approach, where all the modules are combining and verifying the functionality after the completion of individual module testing. In simple words, all the modules of the system are simply put together and tested. This approach is practicable only for very small systems. If once an error is found during the integration testing, it is very difficult to localize the error as the error may potentially belong to any of the modules being integrated. So, debugging errors reported during big bang integration testing are very expensive to fix.

   Advantages:

   - It is convenient for small systems.

Disadvantages:

- There will be quite a lot of delay because you would have to wait for all the modules to be integrated.
- High risk critical modules are not isolated and tested on priority since all modules are tested at once.

2. **Bottom-Up Integration Testing –** In bottom-up testing, each module at lower levels is tested with higher modules until all modules are tested. The primary purpose of this integration testing is each subsystem is to test the interfaces among various modules making up the subsystem. This integration testing uses test drivers to drive and pass appropriate data to the lower-level modules.

Advantages:

- In bottom-up testing, no stubs are required.
- A principal advantage of this integration testing is that several disjoint subsystems can be tested simultaneously.

Disadvantages:

- Driver modules must be produced.
- In this testing, the complexity that occurs when the system is made up of many small subsystems.

3. **Top-Down Integration Testing –** Top-down integration testing technique used to simulate the behavior of the lower-level modules that are not yet integrated. In this integration testing, testing takes place from top to bottom. First high-level modules are tested and then low-level modules and finally integrating the low-level modules to a high level to ensure the system is working as intended.

Advantages:

- Separately debugged module.
- Few or no drivers needed.
- It is more stable and accurate at the aggregate level.

Disadvantages:

- Needs many Stubs.
- Modules at lower level are tested inadequately.

4. **Mixed Integration Testing –** A mixed integration testing is also called sandwiched

integration testing. A mixed integration testing follows a combination of top down and bottom-up testing approaches. In top-down approach, testing can start only after the top-level module have been coded and unit tested. In bottom-up approach, testing can start only after the bottom level modules are ready. This sandwich or mixed approach overcomes this shortcoming of the top-down and bottom-up approaches.

Advantages:

- Mixed approach is useful for very large projects having several sub projects.
- This Sandwich approach overcomes this shortcoming of the top-down and bottom-up approaches.

Disadvantages:

- For mixed integration testing, require very high cost because one part has Top-down approach while another part has bottom-up approach.
- This integration testing cannot be used for smaller system with huge interdependence between different modules.

System Testing

System testing tends to affirm the end-to-end quality of the entire system. System testing is often based on the functional / requirement specification of the system. Non-functional quality attributes, such as reliability, security, and maintainability are also checked. System testing is a process of testing the entire system that is fully functional, in order to ensure the system is bound to all the requirements provided by the client in the form of the functional specification or system specification documentation. In most cases, it is done next to the Integration testing, as this testing should be covering the end-to-end system's actual routine.

This type of testing requires a dedicated Test Plan and other test documentation derived from the system specification document that should cover both software and hardware requirements. By this test, we uncover the errors. It ensures that all the system works as expected. We check System performance and functionality to get a quality product. System testing is nothing but testing the system. This testing checks complete end-to-end scenarios as per the customer's point of view.

Functional and Non-Functional tests also done by System testing. All things are done to maintain trust within the development that the system is defect-free and bug-free. System testing is also intended to test hardware/software requirements specifications. System testing is more of a limited type of testing; it seeks to detect both defects within the "inter-assemblages".

There are three types of system testing.

- Alpha testing is done by the developers who develop the software. This testing is also done by the client or an outsider with the presence of developer or we can say tester.
- Beta testing is done by very few numbers of end users before the delivery, where the change requests are fixed, if the user gives any feedback or reports any type of defect.
- User Acceptance testing is also another level of the system testing process where the system is tested for acceptability. This test evaluates the system's compliance with the client requirements and assess whether it is acceptable for software delivery.
- An error correction may introduce new errors. Therefore, after every round of error-fixing, another testing is carried out, i.e., called regression testing. Regression testing does not belong to either unit testing, integration testing, or system testing, instead, it is a separate dimension to these three forms of testing.

- To test the system as a whole, requirements and expectations should be clear, and the tester needs to understand the real-time usage of the application too.
- Also, most used third-party tools, versions of OSes, flavors and architecture of OSes can affect the system's functionality, performance, security, recoverability or install ability.
- Therefore, while testing the system a clear picture of how the application is going to be used and what kind of issues it can face in real-time can be helpful. In addition to that, a requirements document is as important as understanding the application.
- Clear and updated requirements document can save tester from several misunderstandings, assumptions, and questions.
- In short, a pointed and crisp requirement document with the latest updates along with an understanding of real-time application usage can make ST more fruitful. This testing is done in a planned and systematic manner.

Types of System Testing:

- **Functionality Testing:** To make sure that functionality of the product is working as per the requirements defined, within the capabilities of the system.
- **Recoverability Testing:** To make sure how well the system recovers from various input errors and other failure situations.
- **Interoperability Testing:** To make sure whether the system can operate well with third-party products or not.
- **Performance Testing:** To make sure the system's performance under the various condition, in terms of performance characteristics.
- **Scalability Testing:** To make sure the system's scaling abilities in various terms like user scaling, geographic scaling, and resource scaling.
- **Reliability Testing:** To make sure the system can be operated for a longer duration without developing failures.
- **Regression Testing:** To make sure the system's stability as it passes through an integration of different subsystems and maintenance tasks.
- **Documentation Testing:** To make sure that the system's user guide and other help topics documents are correct and usable.
- **Security Testing:** To make sure that the system does not allow unauthorized access to data and resources.
- **Usability Testing:** To make sure that the system is easy to use, learn and operate.

Regression Testing

The purpose of regression testing is to ensure that bug fixes and new functionality introduced in a software do not adversely affect the unmodified parts of the program. Regression testing is an important activity at both testing and maintenance phases. When a piece of software is modified, it is necessary to ensure that the quality of the software is preserved. To this end, regression testing is to retest the software using the test cases selected from the original test suite.

Regression Testing is usually performed after verification of changes or new functionality. But this is not always the case. For the release that is taking months to complete, regression tests must be incorporated in the daily test cycle. For weekly releases, regression tests can be performed when Functional Testing is over for the changes.

Regression checking is a variation of retest (which is simply to repeat a test). When Retesting, the reason can be anything. Say, you were testing a particular feature and it was the end of the day- you could not finish testing and had to stop the process without deciding if the test passed/failed.

The next day when you come back, you perform the test once more – that means you are repeating a test you performed before. The simple act of repeating a test is a Retest. Regression test at its core is a retest of sorts. It might be code, design, or anything at all that dictates the overall framework of the system.

A Retest that is conducted in this situation to make sure that the said change has not made an impact on anything that was already working before is called the Regression Test.

The most common reason why this might be conducted is because new versions of the code have been created (increase in scope/requirement) or bugs have been fixed.

Automated Regression Testing: Automated Regression Test is a testing area where we can automate most of the testing efforts. We ran all the previously executed test cases on a new build.

This means that we have a test case set available and running these test cases manually is time-consuming. We know the expected results, so automating these test cases is timesaving and is an efficient regression test method. The extent of automation depends upon the number of test cases that are going to remain applicable overtime.

If test cases vary from time to time, the application scope goes on increasing and then automation of regression procedure will be a waste of time.

Most of the Regression testing tools are of record and playback types. You can record the test cases by navigating through the AUT (application under test) and verify whether the expected results are coming or not.

Security Testing

With the rise of cloud-based testing platforms and cyber-attacks, there is a growing concern and need for the security of data being used and stored in software. Security testing is a non-functional software testing technique used to determine if the information and data in a system is protected. The goal is to purposefully find loopholes and security risks in the system that could result in unauthorized access to or the loss of information by probing the application for weaknesses.

The software industry has achieved solid recognition in this age. In recent decades, however, the cyber-world seems to be an even more dominating and driving force which is shaping up the new forms of almost every business.

Web-based ERP systems used today are the best evidence that IT has revolutionized our beloved global village. These days, websites are not only meant for publicity or marketing, but they have evolved into stronger tools to cater to complete business needs.

There are multiple types of this testing method, each of which aimed at verifying six basic principles of security:

1. Integrity
2. Confidentiality
3. Authentication
4. Authorization
5. Availability
6. Non-repudiation

Web-based Payroll systems, shopping malls, Banking, and Stock Trade applications are not only being used by organizations but are also being sold as products today.

This means that online applications have gained the trust of customers and users regarding their vital feature named SECURITY. No doubt, that security factor is of primary value for desktop applications too.

However, when we talk about the web, the importance of security increases exponentially. If an online system cannot protect the transaction data, then no one will ever think of using it.

Examples of Security flaws in an application

- A Student Management System is insecure if the admission branch can edit the data of the ‘Exam’ branch.
- An ERP system is not secure if a DEO (data entry operator) can generate ‘Reports’.

- An online shopping mall has no security if the customer's Credit Card Details are not encrypted.
- A custom software possesses inadequate security if an SQL query retrieves actual passwords of its users.

Desktop And Web Security Testing

A desktop application should be secure not only regarding its access but also with respect to the organization and storage of its data.

Similarly, web applications demand, even more, security with respect to its access, along with data protection. A web developer should make the application immune to SQL Injections, Brute Force Attacks and XSS (cross-site scripting). Similarly, if the web application facilitates remote access points then these must be secure too.

Also, keep in mind that Brute Force Attack is not only related to web applications, but the desktop software is also vulnerable to this.

I hope this foreword is enough and now let me come to the point. Kindly accept my apology if you have so far thought that you are reading about the subject of this article. Though I have briefly explained software security and its major concerns, my topic is "Security Testing".

Security Testing - Techniques:

- Injection
- Broken Authentication and Session Management
- Cross-Site Scripting (XSS)
- Insecure Direct Object References
- Security Misconfiguration
- Sensitive Data Exposure
- Missing Function Level Access Control
- Cross-Site Request Forgery (CSRF)
- Using Components with Known Vulnerabilities
- Unvalidated Redirects and Forwards

Performance Testing

Performance testing is a non-functional testing technique used to determine how an application will behave under various conditions. The goal is to test its responsiveness and stability in real user situations.

Performance Testing Attributes:

- Speed:
  It determines whether the software product responds rapidly.
- Scalability:
  It determines amount of load the software product can handle at a time.
- Stability:
  It determines whether the software product is stable in case of varying workloads.
- Reliability:
  It determines whether the software product is secure or not.

Objective of Performance Testing:

- The objective of performance testing is to eliminate performance congestion.
- It uncovers what is needed to be improved before the product is launched in market.
- The objective of performance testing is to make software rapid.
- The objective of performance testing is to make software stable and reliable.

Performance testing can be broken down into four types:

- Load testing is the process of putting increasing amounts of simulated demand on your software, application, or website to verify whether or not it can handle what it's designed to handle.
- Stress testing takes this a step further and is used to gauge how your software will respond at or beyond its peak load. The goal of stress testing is to overload the application on purpose until it breaks by applying both realistic and unrealistic load scenarios. With stress testing, you'll be able to find the failure point of your piece of software.
- Endurance testing, also known as soak testing, is used to analyze the behavior of an application under a specific amount of simulated load over longer amounts of time. The goal is to understand how your system will behave under sustained use, making it a longer process than load or stress testing (which are designed to end after a few hours). A critical piece of endurance testing is that it helps uncover memory leaks.

- Spike testing is a type of load test used to determine how your software will respond to substantially larger bursts of concurrent user or system activity over varying amounts of time. Ideally, this will help you understand what will happen when the load is suddenly and drastically increased.

id	Test case	Input description	Expected output	Test status
1	Loadind model	Initializing trained model and load it into ON	Loaded model without errors	pass
2	Converting video to frames	Capturing video and converting it into frames	Image frames of captured video stream	pass
3	Recognize hand gesture	Image frame that contains hand object	label	Pass

Table 5.8.1: Verification of test cases of the system

6. CONCLUSION

Sign language is a source of communication for deaf or dumb people. For common people it is difficult to understand sign language. The Real Time Assistive System for Deaf and Dumb Community which provides the opportunity for common people to understand sign language thus bridging the communication gap between the deaf/dumb and the common people.

Real Time Assistive System for Deaf and Dumb Community captures the ASL gestures made by deaf or dumb people in real time and classify these gestures into text. This system can further be enhanced by incorporating dynamic hand gestures as well as facial expressions which make up major part of communication for deaf/dumb people. Most of the times a single gesture conveys the complete message. This requires the system to be trained more rigorously.

7. FUTURE WORK

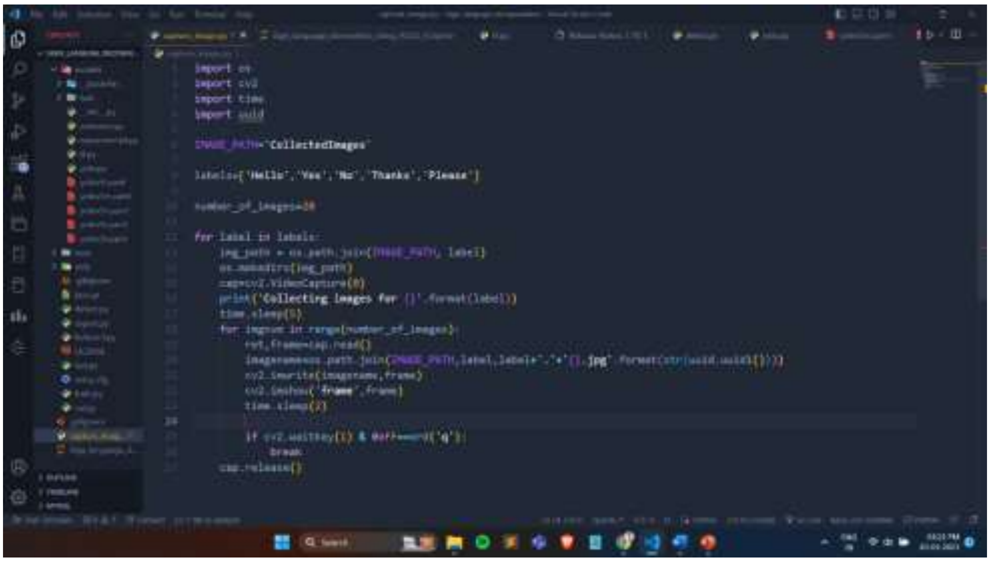
- In the proposed system, we can add features like converting sign done by deaf and dumb people to audio.
- It can also embed our system into various conference apps.
- It can also embed our system into android app so that the user can perform the sign in front of mobile and show output to normal people while communicating.

USER MANUAL

User’s Manual

1. Open VS Code or any editor for opening project.

2.Open the file which consist of project.



3.The user application starts with live camera where he/she can show the specific signs. i.e. Hello, NO, Thanks, Please etc.

'Hello' sign Detected

'NO' Sign Detected

When user stands in front of camera, where he/she can see Realtime functionality.

'Please' Sign Detected

'Thanks' Sign Detected

Here the numbers shown with the sign are the accuracy rate of the respective signs.

REFERENCES

[1] "Design of Sign Language Recognition Using E-CNN" By Citra Suardi and Anik Nur Handayani 2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT) | 978-1-6654-0514-0/20/$31.00 ©2021 IEEE | DOI: 10.1109/EIConCIT50028.2021.9431877

[2] "Convolutional Neural Network Hand Gesture Recognition for American Sign Language" Shruti Chavan, Xinrui Yu and Jafar Saniie 2021 IEEE International Conference on Electro Information Technology (EIT) | 978-1-6654-1846-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/EIT51626.2021.9491897

[3] "Sign Language Alphabet Reorganization Using Convolutional Neural Network" Proceedings of the Fifth International Conference on Intelligent Computing and Control Systems (ICICCS 2021) IEEE Xplore Part Number: CFP21K74-ART

[4] "Research on Communication App for Deaf and Mute People Based on face reorganization Technology" By Yuan Tao and Shihang huao 2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT) | 978-1-7281-9948-1/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICCASIT50869.2020.9368771

[5] "Classification of American Sign Language by Applying a Transfer Learned Deep Convolutional Neural Network" By Md. Mehedi Hasan, Azmain Yakin Srizon 2020 23rd International Conference on Computer and Information Technology (ICCIT), 19-21 December, 2020 978-1-6654-2244-4/20/$31.00 ©2020 IEEE

[6] "Real-Time Sign Language Recognition Based on Video Stream" Proceedings of the 39th Chinese Control Conference July 27-29, 2020, Shenyang, China.

[7] "Hand Gesture Detection based Real-time American Sign Language Letters Recognition using Support Vector Machine". By Xinyun Jiang and Wasim Ahmad. 978-1-7281-3024-8/19/$31.00 ©2019 IEEE DOI 10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00078

[8] "Sign Language Recognition Using Image Based Hand Gesture Recognition Techniques" 2016 Online International Conference on Green Engineering and Technologies (IC-GET) 978-1-5090-4556-3/16/$31.00 ©2016 IEEE

[9] "An Efficient Real-Time Emotion Detection Using Camera and Facial Landmarks" Binh T. Nguyen, Minh H. Trinh, Tan V. Phan and Hien D. Nguyen. 2017 Seventh International Conference on Information Science and Technology (ICIST) 10.1109/ICIST.2017.7926765

[10]"A New Benchmark on American Sign Language Recognition using Convolutional Neural Network" By Md. Moklesur Rahman∗ , Md. Shafiqul Islam 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), 24-25 December, Dhaka 978-1-7281-6099-3/19/$31.00 © 2019 IEEE

[11]“Hand Gesture Recognition for Deaf-Mute using Fuzzy-Neural Network” 2019 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia) 978-1-7281-3336-2/19/$31.00 ©2019 IEEE

[12]“American Sign Language Recognition using Deep Learning and Computer Vision” By Kistij Bantupalli and Ying Xie 2018 IEEE International Conference on Big Data (Big Data) 978-1-5386-5035-6/18/$31.00 ©2018 IEEE

[13] “Hand Gesture Recognition Based on Deep Learning” By Jing-Hao Sun, Ting-Ting Ji and Shu-Bin Zhang 978-1-5386-7302-7/18/$31.00 ©2018 IEEE.

[14] “Sign Language Recognition Based On Hand And Body Skeletal Data” by Dimitrios Konstantinidis, Kosmas Dimitropoulos and Petros Daras, 978-1-5386-6125-3/18/$31.00 ©2018 European Union.

[15] Dr.A.R.Karwankar, Bhushan Bhokse. “Hand Gesture recognition Using Neural Network”.IJISET International Journal of Innovative Science, Engineering and Technology, Vol. 2 Issue 1,January 2015

[16] Joyeeta Singha and Karen Das. “Recognition of Indian Sign Language in Live Video”.International Journal of Computer Applications (0975 – 8887), Volume 70:No. 19, May 2013.

[17] Md. Ashikur Rahman. “Computer Vision Based Human Detection”. International Journal ofEngineering and Information Systems (IJEAIS), Vol. 1 Issue 5:Pages: 62–85, july 2017.

[18] Raman Maini and Dr. Himanshu Aggarwal. “Study and Comparison of Various Image EdgeDetection Techniques”. International Journal of Image Processing (IJIP), Volume (3): issue(1):Pages: 62–85.

[19] SuhasMahishi Dheeraj R Sudheender S Nitin V Pujari Ravikiran J, Kavi Mahesh. “FingerDetection for Sign Language Recognition”. Proceedings of the International MultiConferenceof Engineers and Computer Scientists, Vol. 1:Pages: 18–20, March 2009.

[20] Monica Avlash and Dr.Lakhwinder Kaur. “PERFORMANCES ANALYSIS OFDIFFERENT EDGE DETECTION METHODS ON ROAD IMAGES”. International Journal ofAdvanced Research in Engineering and Applied Sciences, Vol. 2, ISSN: 2278-6252: No. 6, June2013.

[21] Zhao Xu, Xu Baojie and Wu Guoxin, “Canny edge detection based on Open 13th IEEE international conference on Electronic Measurement and instruments (ICEMI), Pages:53-56, doi: 10.1109/ICEMI.2017.8265710, 2017.

[22] Akash ,“ASL Alphabet :Image data set for alphabets in the Americansign language”, Version-1, 2018 [Online].Available: https://www.kaggle.com/grassknoted/asl-alphabet[Accessed:11-Nov2019].

DISSEMINATION OF WORK

- Research Paper Published in Journal:
  1. Paper Name: Real Time Assistive System for Deaf and Dumb Community
     Publication Name: International Journal of Scientific Research in Engineering and Management (IJSREM)
     Author: Ashish Dandade, Pranav Tayade, Rushikesh Patil, Jayant Mitkari
     Issue Number: Volume: 07 Issue: 04 | April - 2023
     DOI: 10.55041/IJSREM19702
     ISSN NO: 2581-9429
     Download Link: https://ijsrem.com/download/real-time-assistive-system-for-deaf-and-dumb-community/

- Participation in Project Competition:
  1. Competition Name: IEEE Technovation 2023
     Organized By: Shri Sant Gajanan Maharaj College of Engineering Shegaon.

IEEE BOMBAY SECTION

Shri Sant Gajanan Maharaj College of Engineering

IEEE BOMBAY SECTION STUDENT ACTIVITIES COMMITTEE

CERTIFICATE

OF PARTICIPATION

THIS CERTIFICATE IS PROUDLY PRESENTED TO

ASHISH DANDADE

FOR ACITIVELY PARTICIPATING IN TECHNOVATION 2023 DIVISION 6 (AKOLA, AMRAVATI, BULDHANA, WASHIM, YAVATMAL) HOSTED BY SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGINEERING(SSGMCE) ON 8TH APRIL, 2023

DR. P.R. WANKHEDE
BRANCH COUNSELOR,
IEEE SB, SSGMCE, SHEGAON

PROF. DATTATRAY SAWANT
CHAIR
STUDENT ACTIVITIES COMMITTEE (SAC)
IEEE BOMBAY SECTION

DR. S. B. SOMANI
PRINCIPAL,
SSGMCE, SHEGAON

IEEE BOMBAY SECTION

Shri Sant Gajanan Maharaj College of Engineering

IEEE BOMBAY SECTION STUDENT ACTIVITIES COMMITTEE

CERTIFICATE

OF PARTICIPATION

THIS CERTIFICATE IS PROUDLY PRESENTED TO

PRANAV TAYADE

FOR ACITIVELY PARTICIPATING IN TECHNOVATION 2023 DIVISION 6 (AKOLA, AMRAVATI, BULDHANA, WASHIM, YAVATMAL) HOSTED BY SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGINEERING(SSGMCE) ON 8TH APRIL, 2023

DR. P.R. WANKHEDE
BRANCH COUNSELOR,
IEEE SB, SSGMCE, SHEGAON

PROF. DATTATRAY SAWANT
CHAIR
STUDENT ACTIVITIES COMMITTEE (SAC)
IEEE BOMBAY SECTION

DR. S. B. SOMANI
PRINCIPAL,
SSGMCE, SHEGAON

IEEE BOMBAY SECTION

Shri Sant Gajanan Maharaj College of Engineering

IEEE BOMBAY SECTION STUDENT ACTIVITIES COMMITTEE

CERTIFICATE

OF PARTICIPATION

THIS CERTIFICATE IS PROUDLY PRESENTED TO

RISHIKESH PATIL

FOR ACITIVELY PARTICIPATING IN TECHNOVATION 2023 DIVISION 6 (AKOLA, AMRAVATI, BULDHANA, WASHIM, YAVATMAL) HOSTED BY SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGINEERING(SSGMCE) ON 8TH APRIL, 2023

DR. P.R. WANKHEDE
BRANCH COUNSELOR,
IEEE SB, SSGMCE, SHEGAON

PROF. DATTATRAY SAWANT
CHAIR
STUDENT ACTIVITIES COMMITTEE (SAC)
IEEE BOMBAY SECTION

DR. S. B. SOMANI
PRINCIPAL,
SSGMCE, SHEGAON

IEEE BOMBAY SECTION

Shri Sant Gajanan Maharaj College of Engineering

IEEE BOMBAY SECTION STUDENT ACTIVITIES COMMITTEE

CERTIFICATE

OF PARTICIPATION

THIS CERTIFICATE IS PROUDLY PRESENTED TO

JAYANT MITKARI

FOR ACITIVELY PARTICIPATING IN TECHNOVATION 2023 DIVISION 6 (AKOLA, AMRAVATI, BULDHANA, WASHIM, YAVATMAL) HOSTED BY SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGINEERING(SSGMCE) ON 8TH APRIL, 2023

DR. P.R. WANKHEDE
BRANCH COUNSELOR,
IEEE SB, SSGMCE, SHEGAON

PROF. DATTATRAY SAWANT
CHAIR
STUDENT ACTIVITIES COMMITTEE (SAC)
IEEE BOMBAY SECTION

DR. S. B. SOMANI
PRINCIPAL,
SSGMCE, SHEGAON

PROJECT GROUP DETAILS

Name: Ashish Murlidhar Dandade
Address: Dongargaon, tehsil. Chikhli, District Buldana.
Mobile No: 7498872932
Email ID: ashishdandade6@gmail.com

Name: Pranav Sanjay Tayade
Address: Rao Nagar, Akola.
Mobile No: 7875065044
Email ID: pranavtayade9058@gmail.com

Name: Rushikesh Raghunath Patil
Address: Gurukul Nagar, Nandurbar.
Mobile No: 8390753634
Email ID: Rushikeshpatil.rp7@gmail.com

Name: Jayant Kishor Mitkari
Address: Saoji Galli, Mehkar, District Buldana.
Mobile No:8551898768
Email Address: jayantmitkari007@gmail.com