

DESIGN AN AI MODEL FOR TRANSLATING HAND GESTURE RECOGNITION TO SPEECH AND TEXT

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology degree in
Information Technology

By

GOKUL KUMAR S (Reg. No 39120033)

JACOB RAJA T (Reg. No 39120041)



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

SATHYABAMA

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

Accredited with Grade “A” by NAAC | 12B Status by UGC | Approved by AICTE

**JEPPIAAR NAGAR, RAJIV GANDHISALAI,
CHENNAI - 600119**

MAY - 202



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **JACOB RAJA T (Reg. No 39120041)**, **GOKUL KUMAR S (Reg. No 39120033)** who carried out the Project Phase-2 entitled "DESIGN AN AI MODEL FOR TRANSLATING HAND GESTURE RECOGNITION TO SPEECH AND TEXT" under my supervision from November 2022 to May 2023.

Internal Guide

Dr. R. JEBERSON RETNA RAJ, M.Tech., Ph.D.,

Head of the Department

Dr. R. SUBHASHINI M.E., Ph.D.,

Submitted for Viva voce Examination held on 10/05/2023

Internal Examiner

External Examiner

DECLARATION

I GOKUL KUMAR S(39120033) hereby declare that the Project Phase-2 Report entitled "DESIGN AN AI MODEL FOR TRANSLATING HAND GESTURE RECOGNITION TO SPEECH AND TEXT" done by me under the guidance of Dr. R. JEBRESON RETNA RAJ, M.Tech.,Ph.D., is submitted in partial fulfillment of the requirements for the award Bachelor of Technology degree in Information Technology.

DATE: 10/05/2023

PLACE: Chennai

Gokul Kumar S.

SIGNATURE OF THE CANDIDATE

NATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

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

I convey my thanks to **Dr. T. SASIKALA M.E., Ph. D, Dean**, School of Computing and **Dr. R. SUBHASHINI M.E., Ph.D.**, Head of the Department, Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. R. JEBERSON RETNA RAJ, M.Tech., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

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

ABSTRACT

Communication provide interaction among the people to exchange the feelings and ideas. The deaf community suffer a lot to interact with the community. Sign language is the way through which the people communicate with each other. In order to provide interaction with normal people there is a system which can convert the sign languages to the understandable form. The purpose of this work is to provide a real-time system which can convert Sign Language (ISL) to the text. Most of the work based on handcrafted feature. In this we are introducing a deep learning approach which can classify the sign using the convolutional neural network. First, This project defines classifier model is designed using the Kera's implementation of convolutional neural network using python. Then we process another real-time system which used skin segmentation to find the Region of Interest in the frame which shows the bounding box. The segmented region is feed to the classifier model to predict the sign. The classifier found to be improving with different background and the angle of the image captured.

TITLE OF CONTENTS

Chapter No	TITLE	Page No.
	ABSTRACT	v
	LIST OF FIGURES	viii
1	INTRODUCTION	
	1.1 Sign language and hand gesture recognition	9
2	LITERATURE SURVEY	
	2.1 Inference of Literature Survey	16
	2.2 Open problem in existing system	16
3	REQUIREMENTS ANALYSIS	
	3.1 Feasibility Studies/Risk Analysis of the Project	17
	3.2 Software Requirements Specification Document	17
	3.3 System Use case	18
4	DESCRIPTION OF PROPOSED SYSTEM	
	4.1 Overall design of proposed system	20
	4.2 Project management plan	21
	4.3 Financial report on estimated costing:	22
	4.4 Software to operation plan	22
5	IMPLEMENTATION DETAILS	
	5.1 Development And Deployment Setup	25
	5.2 Algorithms	25
	5.3 Testing	28
6	RESULT & DISCUSSION	
	6.1 Result	29
7	CONCLUSSION & FUTURE WORK	
	7.1 Conclusion	33
	7.2 Future Work	33

7.3	Research Issues	34
7.4	Implementation Issues	35
REFERENCES		36
APPENDIX		
A.	SCREENSHOT	40
B.	SOURCE CODE	44
C.	RESEARCH PAPER	50

LIST OF FIGURES

Figure No.	TITLE	Page No.
4.1	Architecture of Proposed Model	21
4.2	Architecture Diagram	22
6.1	Hello sign by using CNN algorithm	29
6.2	How are you sign by using CNN algorithm	30
6.3	Get me up sign by using CNN algorithm	30
6.4	Sorry sign by using CNN algorithm	31
6.5	Emergency call sign by using CNN algorithm	31
6.6	Model Accuracy	32
6.7	Model Loss	32

CHAPTER 1

INTRODUCTION

In the world, different languages are used among the people to provide communication while talking about physically impaired people both deaf and dumb community also use different sign languages. The different languages are American Sign Language, Chinese sign language, Indian sign language etc. A real-time system among these community will allow the communication barrier among them. Once it has been converted to using the Computer Vision approach then it can be converted to any language. There are many kinds of research undergone by in this are to build an efficient and accurate system. The previous works done by the researchers using the handcrafted feature but having limitation and used special conditions. The most works are based on pattern recognition, feature extraction based on HOG, SIFT, LBP, etc. But the system using a single feature is not sufficient in most of the cases and the Hybrid approach are introduced to solve this problem. But for a real-time system we need faster methods to solve our problems. Now a days our computers are improved with the speed of processing using parallel implementation. In most of the time our system utilizes a single core for solving a problem. Using the GPU system, the problems can be solved by parallel computing and the number of cores is higher than the CPU system. Using the Deep Learning approach, we can model a self-learning system for our needs. Convolutional neural network is one of the trending deep learning systems which is capable of solving any kind of computer vision problem. In our method we used a Region of interest -Convolutional neural network for the real time Sign Recognition using skin segmentation.

1.1 SIGN LANGUAGE AND HAND GESTURE RECOGNITION

The process of converting the signs and gestures shown by the user into text is called sign language recognition. It bridges the communication gap between people who cannot speak and the general public. Image processing algorithms along with neural networks is used to map the gesture to appropriate text in the training data and hence raw images/videos are converted into respective text that can be read and understood.

Dumb people are usually deprived of normal communication with other people in the society. It has been observed that they find it really difficult at times to interact with normal people with their gestures, as only a very few of those are recognized by most people. Since people with hearing impairment or deaf people cannot talk like normal people so they have to depend on some sort of visual communication in most of the time. Sign Language is the primary means of communication in the deaf and dumb community. As like any other language it has also got grammar and vocabulary but uses visual modality for exchanging information.

The problem arises when dumb or deaf people try to express themselves to other people with the help of these sign language grammars. This is because normal people are usually unaware of these grammars. As a result, it has been seen that communication of a dumb person are only limited within his/her family or the deaf community. The importance of sign language is emphasized by the growing public approval and funds for international project. At this age of Technology, the demand for a computer-based system is highly demanding for the dumb community. However, researchers have been attacking the problem for quite some time now and the results are showing some promise. Interesting technologies are being developed for speech recognition but no real commercial product for sign recognition is actually there in the current market.

The idea is to make computers to understand human language and develop a user-friendly human computer interface (HCI). Making a computer understand speech, facial expressions and human gestures are some steps towards it. Gestures are the non-verbally exchanged information. A person can perform innumerable gestures at a time. Since human gestures are perceived through vision, it is a subject of great interest for computer vision researchers. The project aims to determine human gestures by creating an HCI.

CHAPTER 2

LITERATURE SURVEY

Sergio Escalera et.al implemented the Flex sensors and Arduino UNO to recognize hand gestures and the data acquired from the sensors is analysed using traditional machine-learning algorithms. Compared to these traditional learning models, we developed an adversarial learning approach that performs more accurately. [1]

Koller et.al presented a real-time system based on 3D CNN method for recognizing the sign language. A sign language dataset initially created. For practical applications, RGB cameras are used instead of RGB-D cameras to collect video streams. In order to improve the accuracy of recognition, a 3D-CNN method combined with optical flow processing. [2]

Pavlo Molchanov et.al, presented an Arduino glove-based device is used for the conversion of sign language (ASL) to speech. The sign language glove consists of a simple hand glove fitted with flex sensors, which are used to monitor the amount of bend on the fingers. The analog signals from the sensors are digitally converted and compared with the stored value for recognition of signs. [3]FG Guimaraes et.al, implemented an arduino-based framework used to convert sign language into sound language for bridging the communication gap between deaf mutes and the wider community. The system consists of Flex sensors and micro-controller's boards used to recognize hand/figure motion and translate it to sound. [4]

Priyanka Lokhande et.al, presented a gesture recognition system for translating gesture into speech for other community people. The system employs with smart glove act as a interface for people with hearing or speech disabilities to communicate. The Gesture recognition can be done in two ways, image processing and sensors. The glove uses Flex sensor, Accelerometers, and Touch sensor to detect different gesture in sign language. Using Google Text-Speech, the data will be converted into speech. [5]

Gaolin Fang et.al, presented the device consists of flex sensors for detecting

finger movements and an accelerometer for detecting arm movements. The system designed for identifying the Indian sign language and translate the sign into text and speech. [6]

Houqiang Li et.al, Implements a low-cost control device called Hand Gesture or by using touch sensor for the patients to control the wheelchair and they can survey themselves for their basic needs and they need not to depend on others to guide those patients. The system was developed to indicate the directions for the patients to move on the flat surface and also measure the patient heart beat pulse. [7]

Thomas pock et.al, the work consists of a glove equipped with sensors which senses different sign language gesture, these senses data are fed to Arduino and transfer data to android phone via Bluetooth module, a common android phone is used in this work for sign to voice translation and voice to sign language translation. [8]

Carreira et.al, designed a hardware design of the device. Based on preliminary experimental results on nine simple sign languages, the device demonstrated an average value of 0.74 s to convert a sign language into speech and text, which demonstrate the usefulness of the proposed device. This project describes the development of a sign language translator that converts sign language into speech and text by using wearable device. The glove-based device is able to read the movements of a single arm and five fingers. [9]

Yash and Shah et.al, implemented a glove-based device which will be used for conversion of sign language (ASL) to speech. The basic system consists of two parts; sign language recognition and conversion to text and further to speech. The sign language glove consists of a simple hand glove fitted with flex sensors which is being used for the monitoring the amount of bend on the fingers. Flex means bend, this is the sensors that change the resistance depending on the amount of bend on the sensor. [10]

Alosail et.al, implemented an idea which will assist in removing or atleast reducing this gap between the speech impaired and normal people. The research going on this area mostly focuses on image processing approaches. Our prototype involves Arduino Uno as a microcontroller which is interfaced with flex sensors and accelerometer, gyroscopic sensor for reading the hand gestures. Furthermore, to

perform better execution, we have incorporated an algorithm for better interpretation of data and therefore producing more accurate result. Thereafter, we use python to interface Arduino Uno with a microprocessor and finally converting into speech. [11]

S.S.Shinde et.al, designed the goal of this study is to create a real-time system for hand gesture recognition that recognises hand gestures, hand features such as peak calculation and angle calculation, and then converts gesture images into voice and vice versa. We use a simple night vision webcam with a resolution of 20 megapixels to implement this system. [12]

J.E. Mejia Gamarra et.al, designed a system prototype that is able to automatically recognize sign language to help normal people to communicate more effectively with the diminished people. The Sign to Voice system prototype, S2V, was developed using Feed Forward Neural Network for two-sequence signs detection. Mute people can't speak and normal people don't know the sign language which is used for inter communication between mute people. [13]

Y.Manoharan et.al, designed a machine learning model which successfully and efficiently classifies hand gestures into spellings and sentences. Training and testing data spaces are kept different to yield better result confidence. CNN algorithms are applied aggressively to improve the conversion accuracy. A very important category of computational intelligence with respect to social applicability is the ability to transform any linguistic sign gesture into corresponding language text. This is formally known as Sign Language translation (SLT). [14]

S.Pariselvam et.al, implemented a hand gestures play an important role in human-computer interactions. The term "deep learning" refers to a type of machine learning that uses convolutional neural networks to make recognition easier. Convolutional Neural Networks are multilayer process networks that include an input layer, a convolution layer, a max pooling layer, a fully connected layer, and an output layer. CNN can provide more accurate results when compared to other algorithms. [15]

C.U. Om Kumaret.al, designed the system's goal is convert speech to American Sign Language using an RNN (Recurrent Neural Network) with LSTM (Recurrent Neural Network - Long Short-Term Memory) neural network trained by a

Connectionist Temporal Classification (CTC) neural network is a gesture-based communication to text/speech model that enables a two-way correspondence without the use of an interpreter. [16]

Niranjan panigrahi et.al, presented used a Computer Vision, Machine Learning, and Deep Learning methodologies to adapt ISL recognition in this work. To meet this requirement, the captured image is subjected to a series of pre-processing steps that include various Computer Vision techniques such as gray-scale conversion and thresholding using the OTSU algorithm. To train the system, Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), and pre-trained models, VGG-19 and Inception-V3, employ the Transfer Learning mechanism. [17]

Sunitha K.A et.al, designed a different methods are glove based system, data input device and handicap touch screen methods are used. For text to speech conversion, microcontroller, sensors and measuring devices are used. These methods are expensive and difficult to implement. [18]

D.Manoj Kumar et.al, implemented a first component, Hand Gesture Detector, uses pre-trained models to detect hand gestures. The Image Classifier component categorises and translates the hand signs that are detected. For identified hand signs, the Text and Voice Generator component generates text or audio formatted output. Finally, Text to Sign Converter converts entered English text into animated images based on sign language. Easy talk can detect, translate, and produce relevant outputs with greater accuracy by employing these techniques. This can lead to more effective and efficient communication between the community of people with disabilities and the community of people without disabilities. [19]

B.J.Kranthi et.al, implemented a selection of user-accessible modes, a gesture recognition model based on the SVM algorithm is used. In addition, the system modes send and receive text data from an Android application via a cloud network. The received and sent data will be converted into Morse code pattern and voice form vibrations, respectively. [20]

M. Alonso Salazar Cubas et.al, designed an optimised multilayer artificial neural network approach, the goal of this document is to develop and demonstrate the

operation of a prototype capable of recognising the alphabet of Peruvian Sign Language and translating it into the Spanish language alphabet. It is concluded that this prototype is capable of translating Peruvian Sign Language with a high percentage of accuracy, as well as being viable and cost-effective, so testing the effectiveness of this prototype in multiple people with speech disabilities is recommended. [21]

Almasre, Miada.A et.al, implemented an employs with two sensors: a Microsoft Kinect and a Leap Motion Controller. In this paper, real-time hand gesture letters are recognised with the Kinect SDK Version 2 and LMC using a supervised learning method and natural user interface libraries. [22]

K.V.Fale, Akshay Phalke et.al, designed a portable and smart solution to meet the demand of turning disabled people's gestures into a text and speech synthesis format that is easily understood by others. This Smart Glove also has the ability to manage home appliances and monitor your health. The user will have an application installed that will assist them in switching between several modes of operation of our Smart gloves make life easier and more independent for people with special needs. [23]

S. Vigneshwaran et.al, implemented the Gesture detection mistreatment video and image process is employed for sanctioning the communication between the deaf, dumb and traditional folks. All the obtainable systems aren't moveable and not reasonable to poor folks. This paper introduce, the history of communication technologies that have given higher access to the planet for those sensory disabilities that may be are high communication technologies that improve the communication method of deaf and dumb like persons and create them advanced to speak with the opposite traditional persons.[24]

Choudhary et.al, implemented to convert the human sign language to Voice by the help of some device which could be achieved with the help of flux and arduino combination. There are systems available for sign to speech conversion however none of them provide good user interface. For example if a person who has a disability to speak can stand perform the system and the system converts the human gestures as speech and plays it loud so that the person could actually communicate to a mass

crowd gathering. Also the system could bring high efficiency for users to improve communication. [25]

2.1 INFERENCES FROM LITREATURE SURVEY

The obvious Inference we have arrived from the literature survey that there is a need for this solution which should be easy to access or usable where there is less complexity and an effective solution is proposed to overcome the problem.

A promising technique for human-computer interaction is gesture recognition, which enables users to interact with computers through non-verbal indications like hand gestures.

The accuracy and dependability of gesture recognition systems have been substantially improved by artificial intelligence, increasing their usefulness in a variety of applications.

It can be more natural for users to engage with computers when gesture recognition and speech recognition are combined, and it can also help persons with disabilities communicate.

2.2 OPEN PROBLEMS IN EXISTING SYSTEM

The Open Problem in the Existing System are almost all the solution are based on Glove with sensors like Flex Sensors and accelerometer sensor which becomes difficult for the user to wear and carry all the time and one more drawback on the long term the sensor gets damaged and it's expensive to fix.

To increase accuracy, models for gesture detection to voice conversion largely rely on a wide range of datasets. It is difficult to compile a sizable and varied dataset that accounts for all conceivable gesture and accent variances, though.

In many applications of gesture detection to voice conversion, real-time performance is essential. The user experience can be greatly impacted by the excessive latency that some existing systems may endure.

CHAPTER-3

REQUIREMENT ANALYSIS

3.1 RISK ANALYSIS OF THE PROJECT

In the Existing System, Complex Tasks is carried by Humans and cost becomes high for the user and it also not possible to carry and set it every time, and wired connections will be a disturbance to the user, we are designing this solution to overcome this problem which has capability to detect multiple Sign Language and convert the hand sign language to speech output where wires, or kit is not required and makes the users life and also take less time to configure.

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

The way we create software today has changed dramatically the last 30 years, from the childhood of personal computers in the early 80s to today's powerful devices such as Smartphones, Tablets and PCs. The Internet has also changed the way we use devices and software. We still have traditional desktop applications, but Web Sites, Web Applications and so- called Apps for Smartphones, etc. are dominating the software market today.

We need to find and learn Programming Languages that are suitable for the New Age of Programming. We have today several thousand different Programming Languages today. I guess you will need to learn more than one Programming Language to survive in today's software market.

Software Engineering

Software Engineering is the discipline for creating software applications. A systematic approach to the design, development, testing, and maintenance of software. The main parts or phases in the Software Engineering process are:

- Planning
- Requirements Analysis
- Design
- Implementation

- Testing
- Deployment and Maintenance

Python - IDE

Python is an open source and cross-platform programming language that has become increasingly popular over the last ten years. It was first released in 1991. Latest version is 3.7.0. C Python is the reference implementation of the Python programming language. Written in C, C Python is the default and most widely-used implementation of the language.

Python is a multi-purpose programming language (due to its many extensions), examples are scientific computing and calculations, simulations, web development (using, e.g., the Django Web framework), etc.

Python is an object-oriented programming language (OOP), but you can use Python in basic application without the need to know about or use the object- oriented features in Python.

3.3 SYSTEM USECASE

Use case: Sign-to-speech system for deaf and hard of hearing individuals.

Problem: Deaf and hard of hearing individuals often face communication barriers when interacting with hearing individuals who do not know sign language. This can be especially challenging in situations where the individual needs to communicate with a medical professional, emergency responder, or customer service representative.

Solution: A sign-to-speech system that uses computer vision and natural language processing to translate sign language into spoken words can help to bridge this communication gap.

The system would use a camera or video input to capture the sign language gestures of the user. These gestures would then be analysed by the computer vision algorithm to identify the corresponding words and phrases. The natural language processing component of the system would then convert the identified words and phrases into spoken language output. Python would be used as the primary programming language for the development of the system, leveraging popular

computer vision and natural language processing libraries such as OpenCV. Benefits of the system include improved communication and accessibility for deaf and hard of hearing individuals, increased understanding and awareness of sign language, and greater inclusivity in various settings such as healthcare, education, and customer service.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

4.1 OVERALL DESIGN OF PROPOSED SYSTEM

Sign languages are complex systems that include background noise. The most efficient way to increase classification efficiency and for real-life applications is to perform convolutional neural networks on the image. Data from the image data set, along with a bounding box mapping, is used to train models for most object detection problems. It is expensive to mark the bounding box for each image. As well, we proposed a region of interest predictor based on the segmentation of the skin. From the segmented bounded region, we crop the image and feed to the classifier for prediction.

In order to interact with different electronic devices, we can give hand gestures as input. This way of taking the input can bring a vast change in each one of our life. For the people who are incapable of speaking can utilize the concept of converting hand gestures to speech which will be done by a machine at real time. Very soon the application can be seen in various fields. This greatly inspired us to come to our final and main part of our work, i.e., recognition of hand gestures. All the processed data were trained, validated and tested on different machine learning models. For our classification task, we have used a modern machine learning technique – Adversarial Machine Learning. The main motivation behind using this model is to reach to a maximum possible accuracy even if there is some alteration which has occurred in the data for training or testing phase by external factors. The external factors can be anything or anyone who is trying to change the data and expecting a wrong output, intentionally. Such people may be termed as hackers. The various attacks that are more likely to occur for our deep learning models include fast-gradient sign method (FGSM), basic iterative method (BIM) or momentum iterative method (MIM). The earlier mentioned attacks are some pure form of many gradient- based evading techniques that attackers use for evading the classification model. Adversarial attacks take place when noise is added to the data which in turn, while validating the already trained model may result in the classification of false labels. The proposed model is used to prevent such attacks in order to develop an efficient and CNN model which can rightfully classify

labels despite being feed with noisy data.

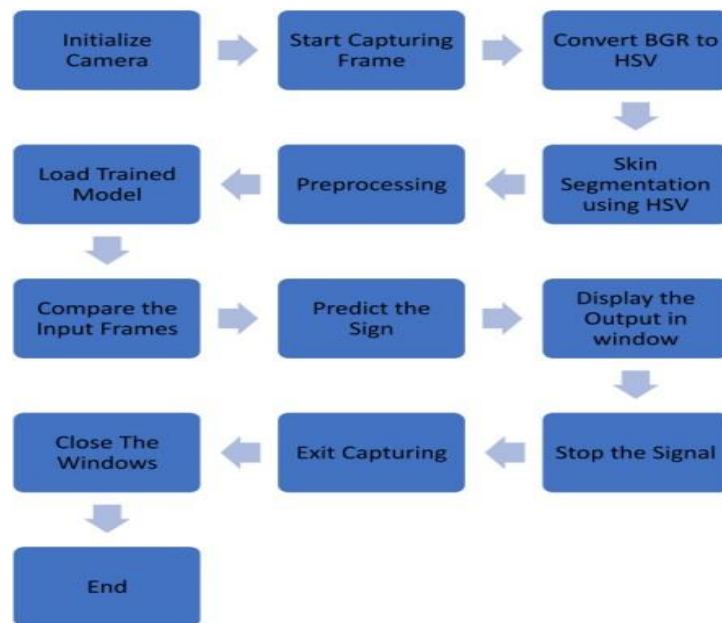


Fig: 4.1 Architecture of Proposed Model

4.2 PROJECT MANAGEMENT PLAN

The sign language is a complex system which includes the background noises. The efficient way is to perform convolutional neural network on the image to increase the efficiency of classification and for real life application. The basic steps involving the proposed system is shown below.

Steps:

- Input the image (video frame).
- Find the hand object.
- Extract the feature.
- Classification and prediction.

Most of the object detection problems train the model using the image data set along with a bounding box mapping. The marking of bounding box for each image is costly. In addition, with that we proposed a region of interest predictor using the skin segmentation. From the segmented bounded region, we crop the image and feed to the classifier for prediction.

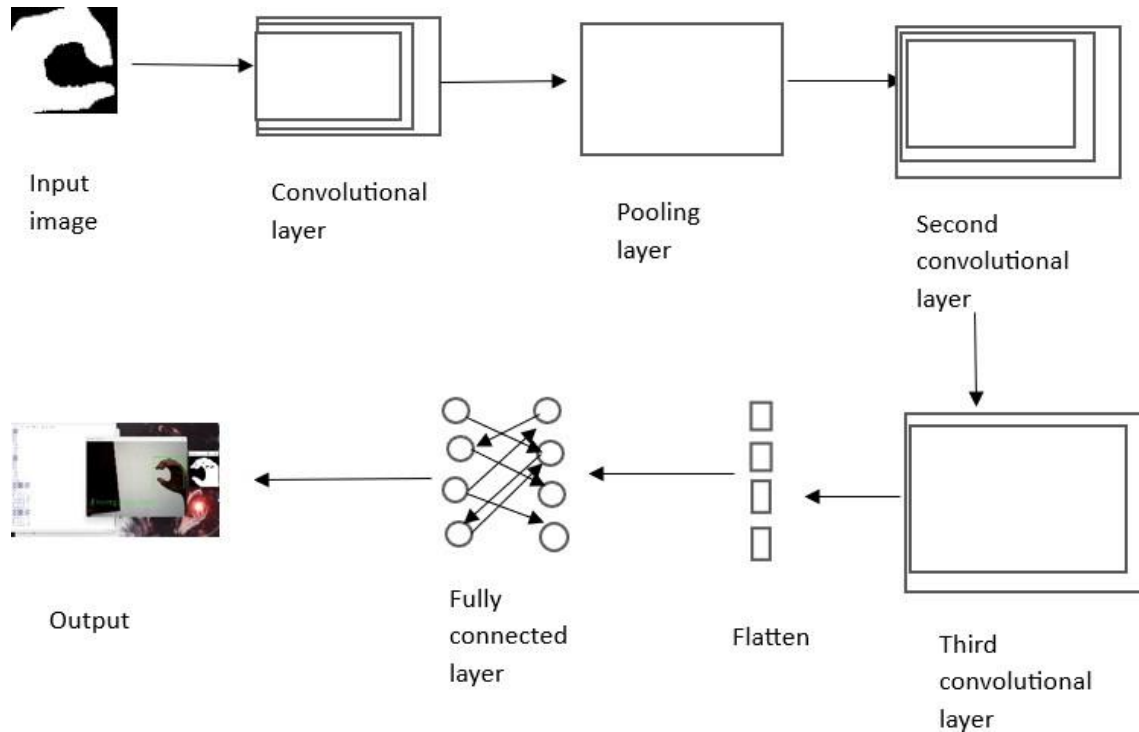


Fig: 4.1 Architecture Diagram

4.3 FINANCIAL REPORT ON ESTIMATED COSTING

There is no Physical Component involved in the Project, you will have only to calculate Appx Man hours involved to develop this concept.

4.4 SOFTWARE TO OPERATION PLAN

INTRODUCTION

This software operation plan outlines the operational requirements for a hypothetical software application. The application is a project management tool designed for small to medium-sized businesses. The plan includes the key objectives, operational processes, and procedures required to ensure the software application functions effectively and efficiently.

Objectives

The primary objectives of the software operation plan are:

- Ensure the software application functions reliably and effectively.

- Ensure the security and privacy of user data.
- Ensure high availability of the software application to minimize downtime.
- Ensure timely and effective support for users.

OPERATIONAL PROCESSES:

The operational processes required to achieve the objectives are:

- **Software Deployment:** The software application will be deployed using a cloud-based platform such as AWS or Azure. The deployment will be managed by a DevOps team, and a continuous integration and deployment pipeline will be established.
- **Security and Privacy:** The software application will be designed to comply with industry-standard security and privacy requirements. This includes the use of encryption, secure authentication, and access control.
- **High Availability:** The software application will be designed to ensure high availability, which will be achieved through the use of redundant infrastructure, load balancing, and failover mechanisms.
- **User Support:** A support team will be established to provide timely and effective support to users. This includes troubleshooting, bug fixes, and feature requests.

Procedures

The procedures required to implement the operational processes are:

- **Software Deployment Procedure:** The deployment procedure will include steps for deploying the software application to the cloud platform, testing the application, and monitoring the application.
- **Security and Privacy Procedure:** The security and privacy procedure will include steps for identifying and mitigating potential security risks, enforcing access control policies, and monitoring access logs.
- **High Availability Procedure:** The high availability procedure will include steps for ensuring redundant infrastructure, load balancing, and failover mechanisms are in place, and monitoring the application for availability.

- User Support Procedure: The user support procedure will include steps for managing user support requests, identifying and resolving issues, and communicating updates to users.

CHAPTER 5

IMPLEMENTATION DETAILS

5.1 DEVELOPMENT AND DEPLOYMENT SETUP

Development Setup

The web application will be developed using the Python programming language. The web application will be built using the Django web framework. The web application will use a database. Git will be used for version control. Python will be used as the IDE for development. The web application will be tested using the unit test module and Selenium for automated testing.

Deployment Setup as Future Scope

The web application will be deployed to a cloud platform such as Amazon Web Services (AWS) or Google Cloud Platform (GCP). The web application will be hosted on a web server such as Apache or Nginx. The deployment process will be managed using a continuous integration and deployment (CI/CD) pipeline. The PostgreSQL database will be hosted on a separate server. The web application will be secured using SSL encryption and access control mechanisms.

5.2 ALGORITHMS

Convolutional neural networks (CNN)

Since Deep Learning is capable of handling large amounts of data, it is proving to be an extremely powerful tool. Pattern recognition has benefited from the use of hidden layers more than traditional methods. The Convolutional Neural Network is one of the most popular deep neural networks. Multiple layers of artificial neurons make up convolutional neural networks. They are mathematical functions that perform a combination of inputs and output an activation value. Artificial neurons are a rough imitation of their biological counterparts. The convolution network generates several activation functions for each layer when you input an image. A basic feature such as a horizontal or diagonal edge is extracted from the first layer. Using this output, the next layer detects more complex features such as corners and combination edges. We can identify more complex features such as objects, faces, and so on, as we move deeper

into the network. A classification layer determines the likelihood that an image belongs to a class based on its activation map generated by the final convolution layer (values from 0 to 1). A ConvoNet, for example, can determine whether a given input image contains a cat, a dog, or a horse by detecting the activation map of the final convolution layer.

Convolution layer

The main building block of a CNN is the convolutional layer. The training process involves learning the parameters of a set of filters (or kernels). Filters usually have a smaller size than the actual image. Each filter creates an activation map by convolution with the image. For convolution the filter slid across the height and width of the image and the dot product between every element of the filter and the input is calculated at every spatial position. The activation map is generated by repeating this process for every element of the input image. The output volume of the convolutional layer is generated by stacking the activation maps of every filter along the depth dimension.

Pooling layer

The Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data by reducing the dimensions. There are two types of pooling average pooling and max pooling. I've only had experience with Max Pooling so far, I haven't faced any difficulties. So, what we do in Max Pooling is we find the maximum value of a pixel from a portion of the image covered by the kernel. Max Pooling also performs as a Noise Suppressant. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel. Average Pooling simply performs dimensionality reduction as a noise suppressing mechanism. Hence, we can say that Max Pooling performs a lot better than Average Pooling.

Second Convolution layer

In a Convolutional Neural Network (CNN), the second convolutional layer applies a set of filters to the output of the first convolutional layer. Each filter is a set of

weights that is convolved with the output of the previous layer, producing a set of activation maps. The second convolutional layer's purpose is to extract more complex and abstract features from the input image. The second layer's filters learn to detect more sophisticated patterns and shapes than the first layers, building on the features discovered in the previous layer.

Overall, the second convolutional layer plays a crucial role in the CNN architecture, allowing the network to learn increasingly complex representations of the input data and improve its ability to perform tasks such as image classification, object detection, and segmentation.

Flattening

There is no flattening layer in the Convolutional Neural Network (CNN) algorithm. Flattening is an operation that is performed on the output of the convolutional and pooling layers, usually before the fully connected layers. The flattening operation converts the 3D tensor output of the convolutional and pooling layers into a 1D vector that can be processed by the fully connected layers. This is accomplished by rearranging the tensor into a long vector by simply concatenating all of the tensor's values into a single dimension. For example, if the output tensor has dimensions of (batch size, height, width, depth), the flattened tensor will have the shape of (batch size, height x width x depth). Overall, flattening is an important operation in the CNN algorithm that enables the extraction of important features from input images and their transformation into a format suitable for classification or regression tasks.

Fully Connected Layer

A Fully Connected (FC) layer in a Convolutional Neural Network (CNN) is a neural network layer where every neuron in the layer is connected to every neuron in the previous layer. In contrast to the convolutional and pooling layers, which operate on spatial features, the FC layer operates on the flattened output of the preceding layers. The purpose of the FC layer is to use the extracted features from the previous layers to make a prediction about the input image. The FC layer is typically placed at the end of the CNN architecture, after the convolutional and pooling layers have extracted important features from the input data. The output of the FC layer is a probability distribution over the possible classes, which is then used to make a

prediction. Overall, the Fully Connected layer plays a critical role in the CNN algorithm, allowing the network to use the extracted features from the previous layers to make a prediction about the input data.

5.3 TESTING

- **Unit Testing:** Unit testing involves testing individual components of the system, such as the data pre-processing pipeline, the CNN model, and the user interface. This ensures that each component works correctly and can be integrated into the system without issues.
- **Integration Testing:** Integration testing involves testing the interactions between different components of the system, such as the CNN model and the user interface, or the backend server and the database. This ensures that the components work together as intended and that the system as a whole meets the functional requirements.
- **Performance Testing:** Performance testing involves testing the system under different load and stress conditions to ensure that it meets the required performance standards. This includes testing the system for real-time performance, scalability, and resource utilization.
- **Validation Testing:** Validation testing involves testing the accuracy and generalization of the system on a validation or test dataset. This ensures that the system can recognize sign language gestures accurately and that it can generalize to new sign language data.
- **User Acceptance Testing:** User acceptance testing involves testing the system with real users to ensure that it meets their needs and expectations. This includes testing the user interface, ease of use, and overall user experience.

CHAPTER 6

RESULTS & DISCUSSION

The entire code for the experiment was developed using Keras, a Python framework which is used commonly for developing deep learning models. Additionally, other libraries used consisted of Scikit-learn, pandas and NumPy. The experiment required the use of a computer server with dedicated Graphical Processing Unit (GPU) as a result of which a machine with AMD Ryzen 5 3550H processor and Nvidia GeForce GTX 1050 graphics card was used. The Adversarial Learning was found to have outperformed standard classifiers. The comparisons were done by running all the machine learning algorithms on the dataset obtained from our initial work. We have tried our best to keep all the necessary conditions as similar as possible in order to have an ideal comparison of all of the models being compared. The below table shows the performance matrix of CNN used before the application of Adversarial Learning on our dataset.

RESULT OF SIGN LANGUAGE

Sign 1

The following image is the result of hello sign, which has both text and audio message to convey special people

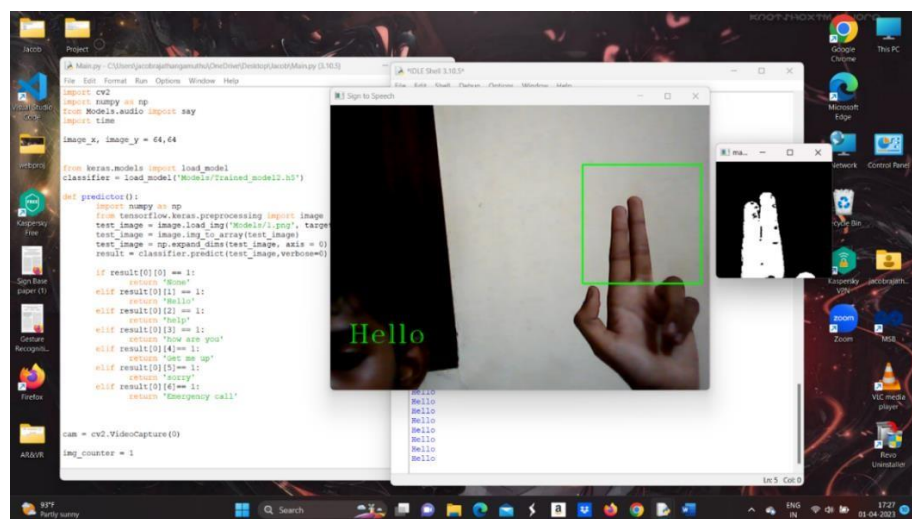


Fig:6.1 hello

Sign 2

As the sign says how are you with both text and audio message for who are not able to dumb people.

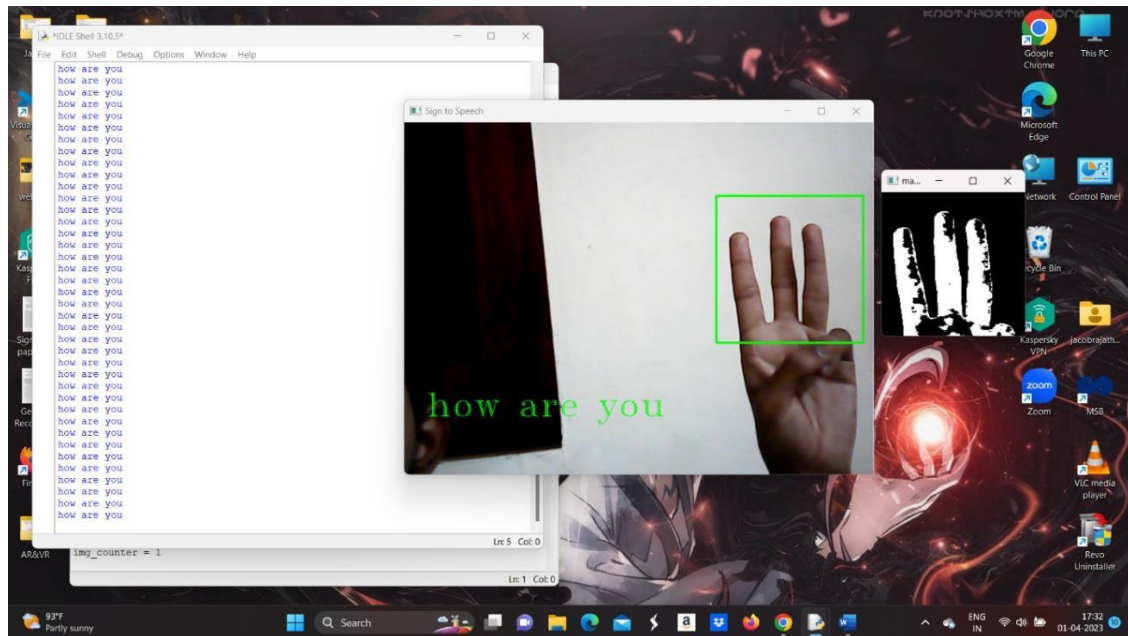


Fig: 6.2 how are you

Sign 3

By this sign language we can indicate the special people, to get me up with both text and audio message.

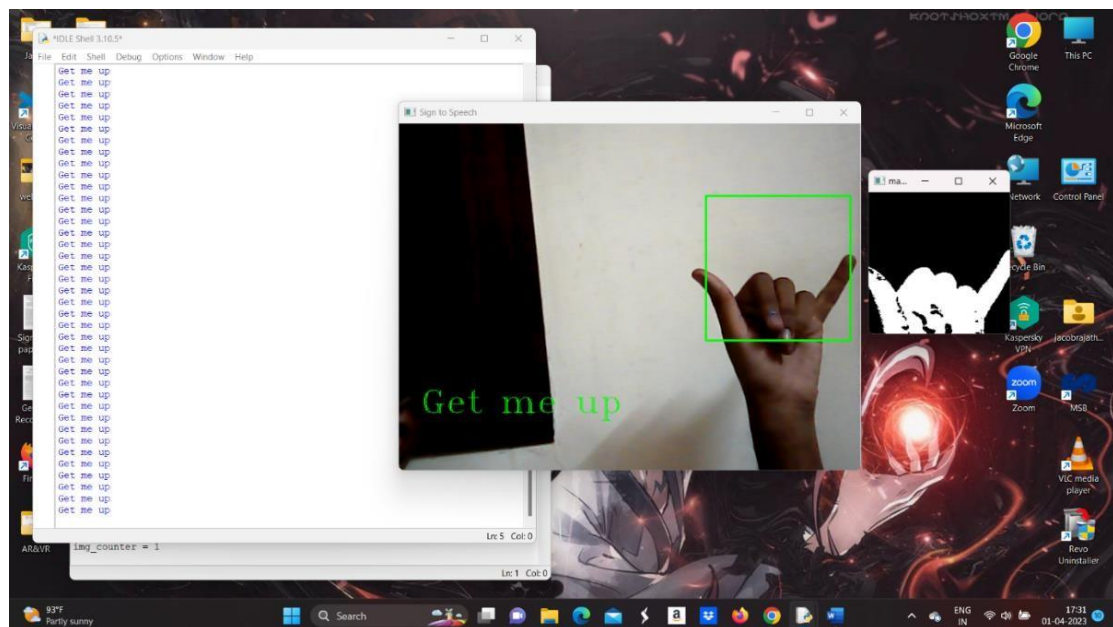


Fig: 6.3 Get me up

Sign 4

By this the word sorry can be easily conveyed to the special people with both txt and audio.

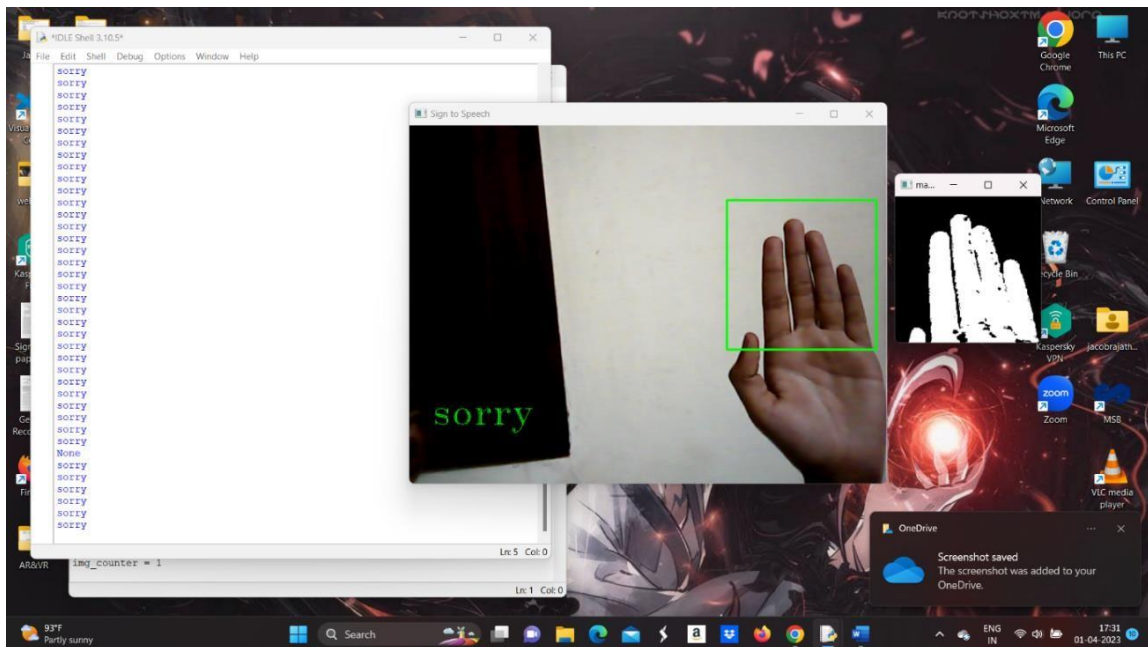


Fig: 6.4 sorry

Sign 5

For emergency purpose this sign can be used for emergency call with both txt and audio.

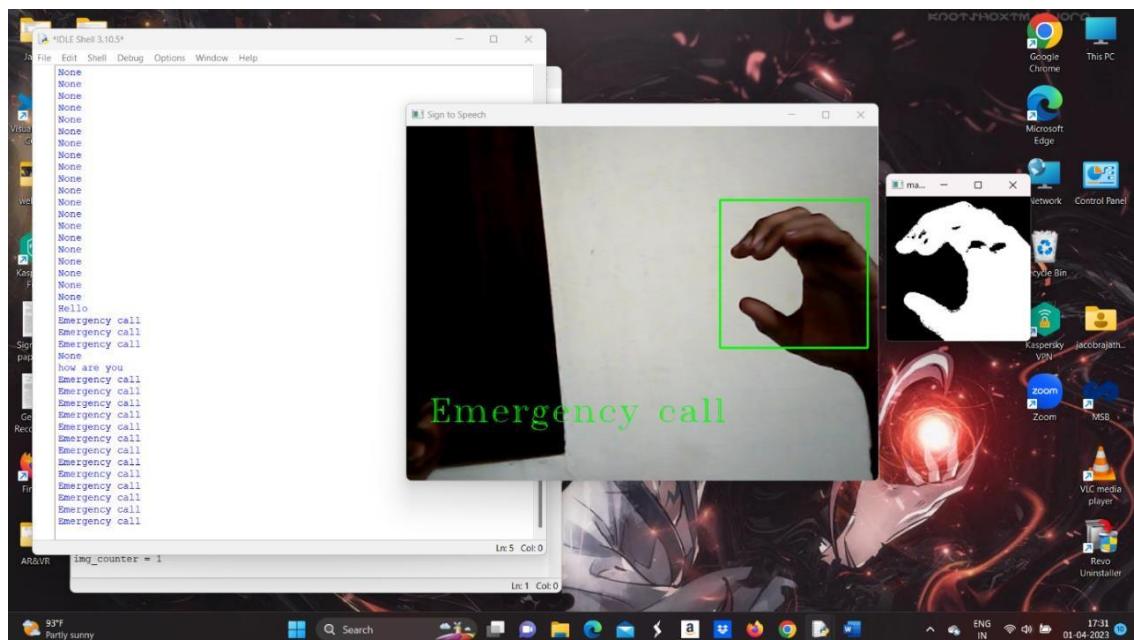


Fig: 6.5 emergency call

Model accuracy

The following graph is the result for the artificial intelligence based on gesture recognition to speech conversion for model accuracy.

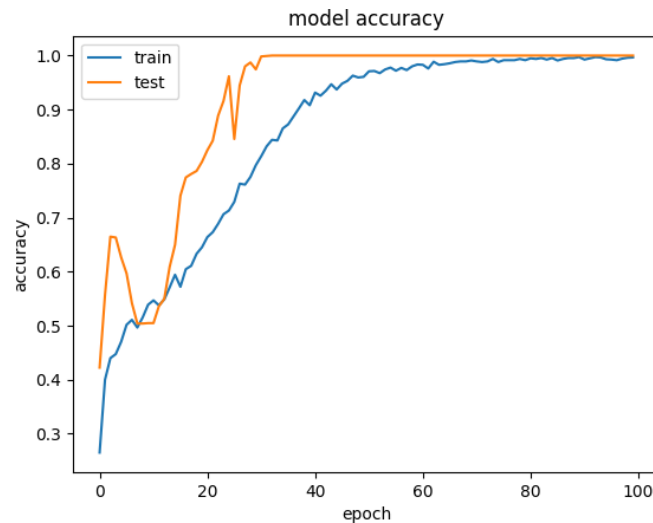


Fig: 6.6 Model accuracy

Model Loss

The following graph is the result for the artificial intelligence based on gesture recognition to speech conversion, which indicates model loss.

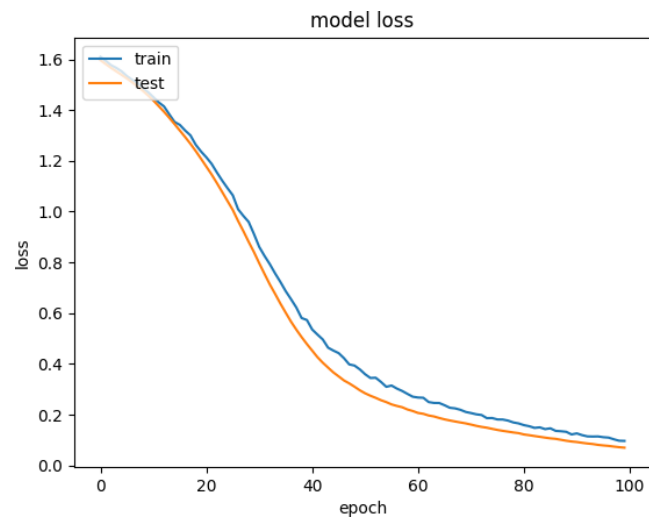


Fig: 6.7 Model Loss

CHAPTER 7

CONCLUSION

7.1 CONCLUSION

A real-time sign language system based on RGB video streams and 3D-CNN is proposed, and a Chinese sign language dataset is created, which provides the basis for a set of sign language recognition. We have shown that sign language recognition can be achieved by combining RGB video streams with TV-L1 optical flow calculations and extracting features through 3D-CNN. We have also shown the creation of a real-time sign language system. Motion detection and human detection are control centers of real-time systems. Frame difference method is a simple and effective motion detection algorithm, which has little effect on the real-time performance of the system. YOLO and Faster R-CNN were proposed for hand and head detection at the same time. We evaluated both networks and finally chose YOLO with better comprehensive performance. According to the evaluation of the experimental results, it is difficult for optical flow to capture subtle movements of the hand. This makes it difficult for 3D-CNN to extract complete motion information. In the future, we can add skin detection and human skeleton detection to improve the accuracy of motion detection. More effective feature detection and tracking from hand movement is also an interesting topic for further explorations.

7.2 FUTURE WORK

Gesture recognition to voice conversion is an intriguing area of study with many potential uses in healthcare, education, entertainment, and other fields. Here are some suggestions for future work in this area:

Improve accuracy: The ability to recognize gestures and translate them into speech can be improved by improving the accuracy of the gesture recognition system. Techniques like deep learning, machine learning, or advancing sensor technology can be used to accomplish this.

The majority of gesture recognition to voice conversion programmers now only have a small vocabulary. Vocabulary expansion can enable more nuanced and varied

communication.

Real-time recognition: The system's usability can be enhanced by real-time gesture recognition and speech conversion, which will make it more effective and user-friendly.

Integration of several modalities: The accuracy and efficiency of the gesture detection system can be increased by integrating additional modalities such as facial expressions, eye gaze, and body position.

Flexibility: The system should be able to adjust to different users' unique gestures and communication preferences. Machine learning algorithms that are tailored to the individual can do this.

Support for different languages: By including support for multiple languages, the system may be more widely usable.

A more user-friendly and intuitive user interface can enhance the user experience and make the system more appealing and simple to use.

Mobile applications: Creating a mobile app that converts gestures into speech can make the system more accessible and enable users to communicate while on the go.

7.3 RESEARCH ISSUES

One of the main challenges in developing CNN-based sign to speech recognition systems is the limited availability of sign language data. The size and quality of the dataset significantly affect the accuracy and generalization of the model. Researchers are exploring new approaches to collect more sign language data to train and test CNN models, such as crowdsourcing, synthetic data generation, and transfer learning. Sign language involves complex and dynamic hand and body gestures, which require high-level understanding of the context and grammar of sign language. CNN models are still limited in their ability to capture the temporal and spatial features of sign language gestures. Researchers are exploring new architectures, such as 3D CNNs and recurrent neural networks (RNNs) that can better capture the temporal and spatial features of sign language. CNN models trained on one sign language dataset may not perform well on a different sign language dataset or in a different sign language environment. Researchers are exploring domain adaptation techniques to improve the generalization of CNN models for sign to speech recognition. These techniques include

transfer learning, fine-tuning, and adversarial learning. CNN models trained on one sign language dataset may not perform well on a different sign language dataset or in a different sign language environment. Researchers are exploring domain adaptation techniques to improve the generalization of CNN models for sign to speech recognition. These techniques include transfer learning, fine-tuning, and adversarial learning.

7.4 IMPLEMENTATION ISSUES

Pre-processing the sign language data is crucial for improving the accuracy of the CNN model. This includes data cleaning, normalization, and augmentation. Data augmentation techniques such as rotation, translation, and scaling can be used to increase the size of the dataset and improve the robustness of the model. can be used to increase the size of the dataset and improve the robustness of the model. The training of the CNN model can take a significant amount of time and requires high computational resources. It is important to use a powerful GPU and distributed training techniques such as data parallelism and model parallelism to speed up the training process. Evaluating the performance of the CNN model is crucial for ensuring the accuracy and generalization of the model. The evaluation metrics such as accuracy, precision, recall, and F1 score should be used to assess the performance of the model on the validation and test sets. Deploying the sign to speech recognition system involves integrating the CNN model with the user interface and the backend server. The user interface should be designed to enable easy and intuitive communication between the sign language users and non-sign language users. The backend server should be optimized for real-time performance and scalability.

REFERENCES

- [1] Sergio Escalera, Xavier Baró, Jordi González, Miguel A. Bautista, & Isabelle Guyon. ChaLearn looking at people challenge 2014: Dataset and results. Workshop at the European Conference on Computer Vision. Springer International Publishing, 2019: 459–473.
- [2] Koller, Oscar, Forster, Jens, Ney, Hermann. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers [J]. Computer Vision & Image Understanding, 2020, 141:108-125.
- [3] Pavlo Molchanov, Xiaodong Yang, Shalini Gupta et al. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural networks[C]// IEEE Conference on Computer Vision and Pattern Recognition, 2019.
- [4] SGM Almeida, FG Guimarães, JA Ramírez. Feature extraction in Brazilian Sign Language Recognition based on phonological structure and using RGB-D sensors [J]. Expert Systems with Applications, 2021, 41(16):7259-7271.
- [5] Priyanka Lokhande, Riya Prajapati, Sandeep Pansare. Data gloves for sign language recognition system [J]. International Journal of Computer Applications, 2015, 11-14.
- [6] Wen Gao, Gaolin Fang, Debin Zhao, Yiqiang Chen. A Chinese sign language recognition system based on SOFM/SRN/HMM [J]. Pattern Recognition, 2020, 37(12):2389-2402.
- [7] Jie Huang, Wengang Zhou, Houqiang Li, Weiping Li. Sign Language Recognition using 3D convolutional neural networks[C]// IEEE International Conference on Multimedia and Expo (ICME), 2020.
- [8] Christopher Zach, Thomas Pock, Horst Bischof. A Duality Based Approach for Realtime TV-L1 Optical Flow[C] Proceedings of the 29th DAGM conference on Pattern recognition. 2019.
- [9] Carreira, Joao, Zisserman, Andrew. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset [J]. IEEE Conference on Computer Vision and Pattern Recognition, 2021.
- [10] Jhunjhunwala, Yash & Shah, Pooja & Patil, Pradnya & Waykule, Jyoti. (2017). SIGN LANGUAGE TO SPEECH CONVERSION USING ARDUINO.

- [11] Alosail, H. Aldolah, L. Alabdulwahab, A. Bashar and M. Khan, "Smart Glove for Bi-lingual Sign Language Recognition using Machine Learning," 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 2023, pp. 409-415, doi: 10.1109/IDCIoT56793.2023.10053470.
- [12] S. S. Shinde, R. M. Autee and V. K. Bhosale, "Real time two way communication approach for hearing impaired and dumb person based on image processing," 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Chennai, India, 2016, pp. 1-5, doi: 10.1109/ICCIC.2016.7919572.
- [13] J. E. Mejía Gamarra, M. Alonso Salazar Cubas, J. D. Sosa Silupú and C. Enrique Córdova Chirinos, "Prototype for Peruvian Sign Language translation based on an artificial neural network approach," 2020 IEEE XXVII International Conference on Electronics, Electrical Engineering and Computing (INTERCON), Lima, Peru, 2020, pp. 1-4, doi: 10.1109/INTERCON50315.2020.9220257.
- [14] Y. Manoharan, S. Saxena and D. R, "A Vision-based Smart Human Computer Interaction System for Hand-gestures Recognition," 2022 1st International Conference on Computational Science and Technology (ICCST), CHENNAI, India, 2022, pp. 321-324, doi: 10.1109/ICCST55948.2022.10040464.
- [15] S. Pariselvam, D. N., D. S. and S. B., "An Interaction System Using Speech and Gesture Based on CNN," 2020 International Conference on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, 2020, pp. 1-5, doi: 10.1109/ICSCAN49426.2020.9262343.
- [16] C. U. Om Kumar, K. P. K. Devan, P. Renukadevi, V. Balaji, A. Srinivas and R. Krithiga, "Real Time Detection and Conversion of Gestures to Text and Speech to Sign System," 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2022, pp. 73-78, doi: 10.1109/ICESC54411.2022.9885562.
- [17] N. Panigrahi, "Artificial Intelligence based Indian Sign Language Recognition with Accelerated Performance under HPC Environment," 2022 OITS

International Conference on Information Technology (OCIT), Bhubaneswar, India, 2022, pp. 228-232, doi: 10.1109/OCIT56763.2022.00051.

- [18] Sunitha K. A, Anitha Saraswathi.P, Aarthi.M, Jayapriya. K, Lingam Sunny, "Deaf Mute Communication Interpreter- A Review", *International Journal of Applied Engineering Research*, Volume 11, pp 290-296, 2016.
- [19] D. Manoj Kumar, K. Bavanraj, S. Thavananthan, G. M. A. S. Bastiansz, S. M. B. Harshanath and J. Alosious, "EasyTalk: A Translator for Sri Lankan Sign Language using Machine Learning and Artificial Intelligence," 2020 2nd International Conference on Advancements in Computing (ICAC), Malabe, Sri Lanka, 2020, pp. 506-511, doi: 10.1109/ICAC51239.2020.9357154.
- [20] B. J. Kranthi, G. Suhas, K. B. Varma and G. P. Reddy, "A two-way communication system with Morse code medium for people with multiple disabilities," 2020 IEEE 7th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Prayagraj, India, 2020, pp. 1-6, doi: 10.1109/UPCON50219.2020.9376479.
- [21] J. E. Mejía Gamarra, M. Alonso Salazar Cubas, J. D. Sosa Silupú and C. Enrique Córdova Chirinos, "Prototype for Peruvian Sign Language translation based on an artificial neural network approach," 2020 IEEE XXVII International Conference on Electronics, Electrical Engineering and Computing (INTERCON), Lima, Peru, 2020, pp. 1-4, doi: 10.1109/INTERCON50315.2020.9220257.
- [22] Almasre, Miada A., and Hana Al-Nuaim. "A Real-Time Letter Recognition Model for Arabic Sign Language Using Kinect and Leap Motion Controller v2." *International Journal of Advanced Engineering, Management and Science*, vol. 2, no. 5, May. 2016.
- [23] K.V.Fale, Akshay Phalke, Pratik Choudhari and Pradeep Jadhav, "SMART GLOVE: Gesture Vocalizer for Deaf and Dumb People", *International Journal of Innovative research in Computer and Communication Engineering*, Vol.\$, Issue \$, pp. 6800-6806, April 2016.
- [24] S. Vigneshwaran, M. Shifa Fathima, V. Vijay Sagar and R. Sree Arshika, "Hand Gesture Recognition and Voice Conversion System for Dump People," *2019 5th International Conference on Advanced Computing & Communication Systems*

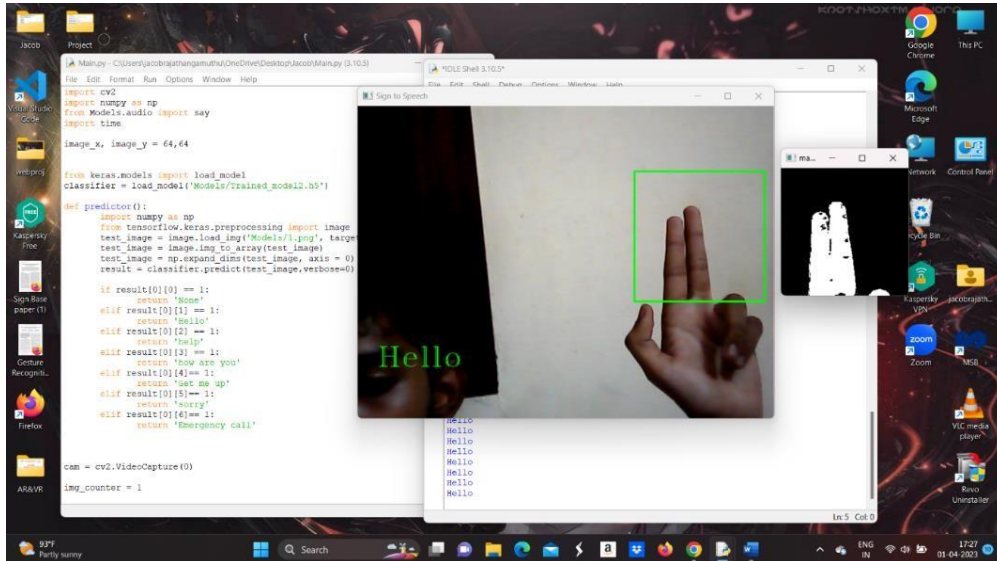
(ICACCS), Coimbatore, India, 2019, pp. 762-765, doi: 10.1109/ICACCS.2019.8728538.

- [25] Choudhary, Dhirendra Kumar and Singh, Rameet and Kamthania, Deepali, Sign Language Recognition System (April 22, 2021). Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2021

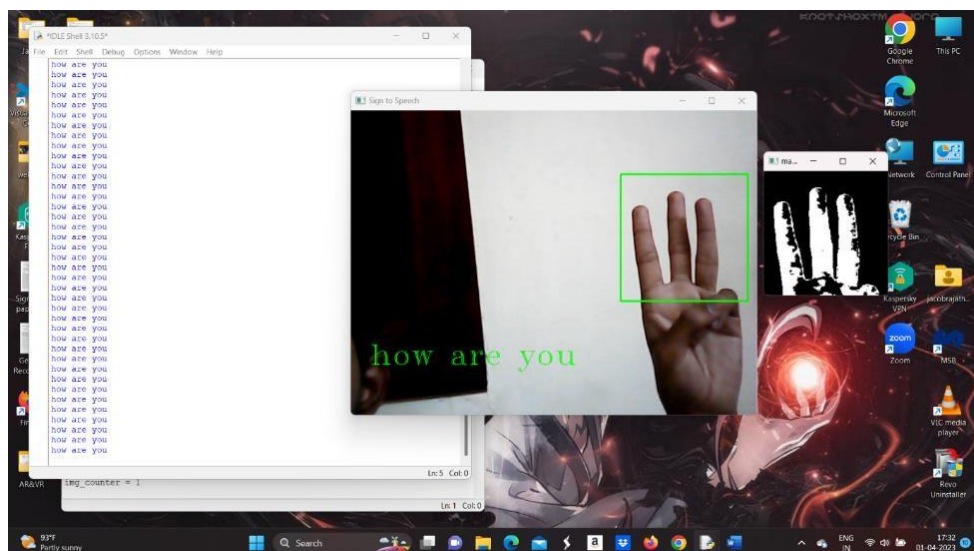
APPENDIX

A. SCREENSHOT

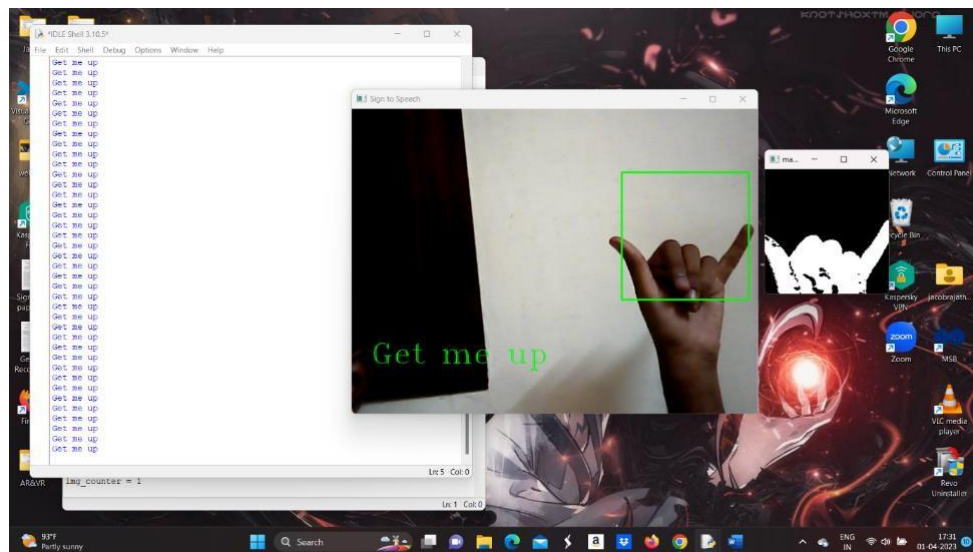
a) Hello



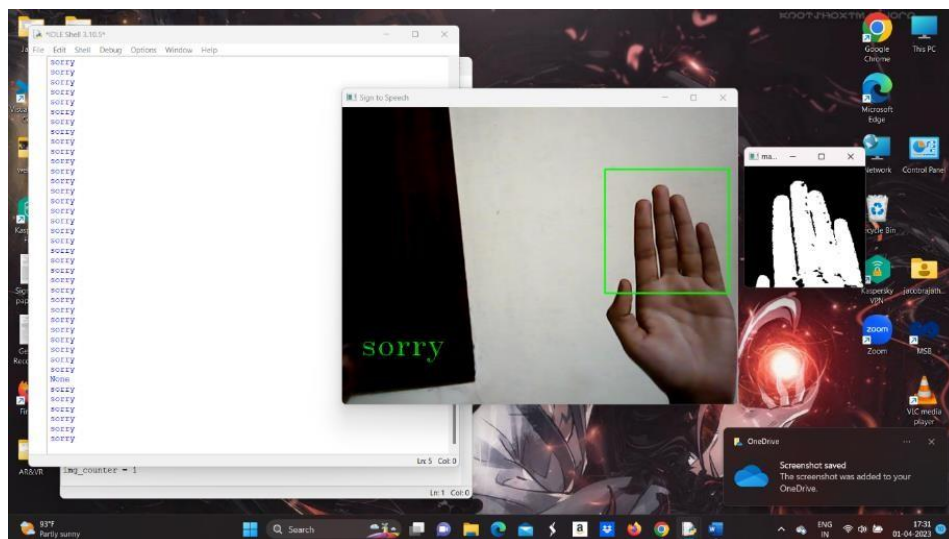
b) How are you



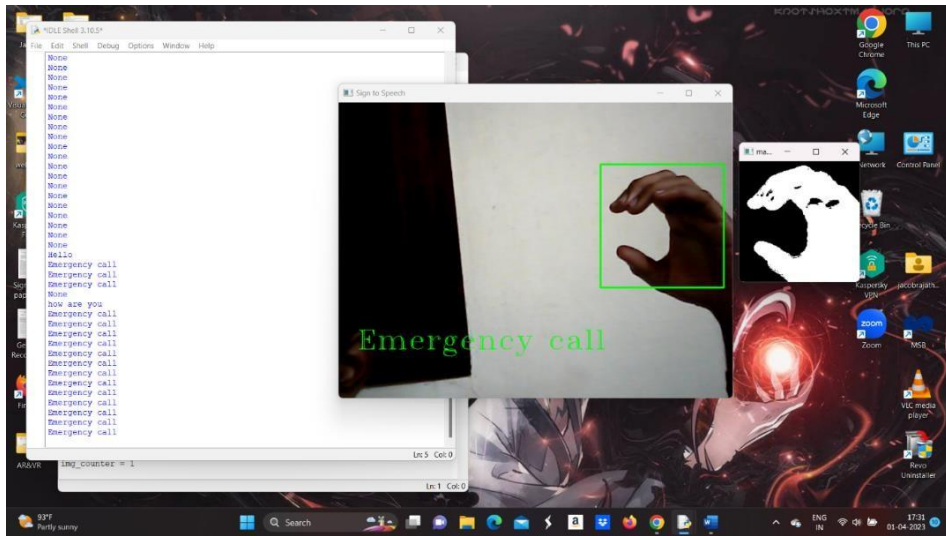
c) Get me up



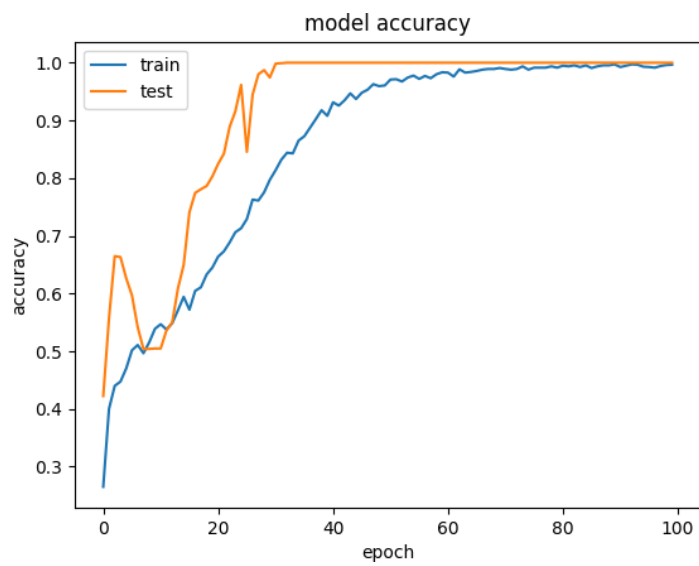
d) Sorry



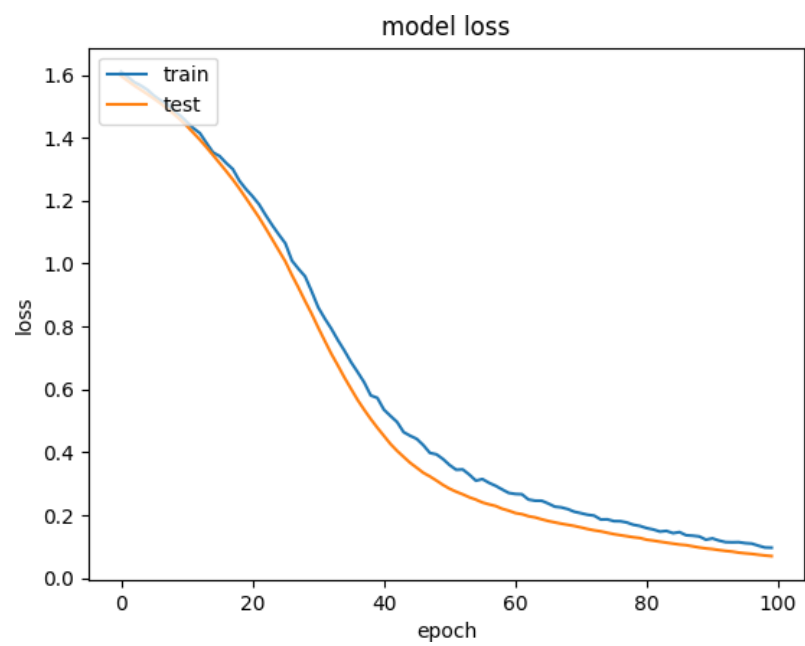
e) Emergency Call



f) Model Accuracy



g) Model Loss



B. SOURCE CODE

CAPTURING DATASET

```
import cv2
import time
import numpy as np
import os

def nothing(x):
    pass

image_x, image_y = 64, 64

def create_folder(folder_name):
    if not os.path.exists('./mydata/training_set/' + folder_name):
        os.mkdir('./mydata/training_set/' + folder_name)
    if not os.path.exists('./mydata/test_set/' + folder_name):
        os.mkdir('./mydata/test_set/' + folder_name)

def capture_images(ges_name):
    create_folder(str(ges_name))

    cam = cv2.VideoCapture(0)

    cv2.namedWindow("test")

    img_counter = 0
    t_counter = 1
    training_set_image_name = 1
    test_set_image_name = 1
    listImage = [1,2,3,4,5]

    cv2.namedWindow("Trackbars")

    cv2.createTrackbar("L - H", "Trackbars", 0, 255, nothing)
    cv2.createTrackbar("L - S", "Trackbars", 70, 255, nothing)
    cv2.createTrackbar("L - V", "Trackbars", 0, 255, nothing)
```

```

cv2.createTrackbar("U - H", "Trackbars", 179, 255, nothing)
cv2.createTrackbar("U - S", "Trackbars", 255, 255, nothing)
cv2.createTrackbar("U - V", "Trackbars", 255, 255, nothing)

for loop in listImage:
    while True:

        ret, frame = cam.read()
        frame = cv2.flip(frame, 1)

        l_h = cv2.getTrackbarPos("L - H", "Trackbars")
        l_s = cv2.getTrackbarPos("L - S", "Trackbars")
        l_v = cv2.getTrackbarPos("L - V", "Trackbars")
        u_h = cv2.getTrackbarPos("U - H", "Trackbars")
        u_s = cv2.getTrackbarPos("U - S", "Trackbars")
        u_v = cv2.getTrackbarPos("U - V", "Trackbars")

        img = cv2.rectangle(frame, (425, 100), (625, 300), (0, 255, 0),
thickness=2, lineType=8, shift=0)
        #img = cv2.rectangle(frame, (400,50),(625,355), (0,255,0), thickness=2,
lineType=8, shift=0)

        lower_skin_color = np.array([l_h, l_s, l_v])
        upper_skin_color = np.array([u_h, u_s, u_v])
        imcrop = img[102:298, 427:623]
9+
        #imcrop = img[52:351, 402:623]
        hsv = cv2.cvtColor(imcrop, cv2.COLOR_BGR2HSV)
        mask = cv2.inRange(hsv, lower_skin_color, upper_skin_color)

        result = cv2.bitwise_and(imcrop, imcrop, mask=mask)

        cv2.putText(frame, str(img_counter), (30, 400),
cv2.FONT_HERSHEY_TRIPLEX, 1.5, (127, 127, 255))
        cv2.imshow("test", frame)
        cv2.imshow("mask", mask)
        #cv2.imshow("result", result)

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

            if t_counter <= 350:
                img_name = "./mydata/training_set/" + str(ges_name) +
"/{}.png".format(training_set_image_name)
                save_img = cv2.resize(mask, (image_x, image_y))

```

```

        cv2.imwrite(img_name, save_img)
        print("{} written!".format(img_name))
        training_set_image_name += 1

    if t_counter > 350 and t_counter <= 610:
        img_name = "./mydata/test_set/" + str(ges_name) +
"/{}.png".format(test_set_image_name)
        save_img = cv2.resize(mask, (image_x, image_y))
        cv2.imwrite(img_name, save_img)
        print("{} written!".format(img_name))
        test_set_image_name += 1
        if test_set_image_name > 250:
            break

    t_counter += 1
    if t_counter >= 610:
        t_counter = 1
        img_counter += 1

elif cv2.waitKey(1) == 27:
    break

if test_set_image_name > 250:
    break

cam.release()
cv2.destroyAllWindows()

ges_name = input("Enter data id: ")
capture_images(ges_name)

```

Main source code

```

import cv2

import numpy as np

from Models.audio import say

import time

```

```
image_x, image_y = 64,64
```

```
from keras.models import load_model
```

```
classifier = load_model('Models/Trained_model2.h5')
```

```
def predictor():
```

```
    import numpy as np
```

```
    from tensorflow.keras.preprocessing import image
```

```
    test_image = image.load_img('Models/1.png', target_size=(64, 64))
```

```
    test_image = image.img_to_array(test_image)
```

```
    test_image = np.expand_dims(test_image, axis = 0)
```

```
    result = classifier.predict(test_image,verbose=0)
```

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

```
        return 'None'
```

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

```
        return 'Hello'
```

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

```
        return 'help'
```

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

```
        return 'how are you'
```

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

```
        return 'Get me up'
```

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

```
        return 'sorry'
```

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

```
        return 'Emergency call'
```

```
cam = cv2.VideoCapture(1)
```

```
img_counter = 1
```

```
img_text = "
```

```
found = set()
```

```
while True:
```

```
    ret, frame = cam.read()
```

```
    frame = cv2.flip(frame,1)
```

```
    l_h=0
```

```
    l_s=70
```

```
    l_v=0
```

```
    u_h = 179
```

```
    u_s = 255
```

```
    u_v = 255
```

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

```
    lower_hsv = np.array([l_h, l_s, l_v])
```

```
    upper_hsv = np.array([u_h, u_s, u_v])
```

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

```
    hsv = cv2.cvtColor(imcrop, cv2.COLOR_BGR2HSV)
```

```
    mask = cv2.inRange(hsv, lower_hsv, upper_hsv)
```

```

img_name = "Models/1.png"
save_img = cv2.resize(mask, (image_x, image_y))
cv2.imwrite(img_name, save_img)
img_text = predictor()
print(img_text)
cv2.putText(frame, img_text, (30, 400), cv2.FONT_HERSHEY_TRIPLEX,
1.5, (0, 255, 0))
cv2.imshow("Sign to Speech", frame)
cv2.imshow("mask", mask)
if(str(img_text) != 'None'):
    if str(img_text) not in found :
        say(str(img_text))
        found.add(str(img_text))
        #time.sleep(2)
    else:
        found.clear()

if cv2.waitKey(1) == 27:
    break

cam.release()
cv2.destroyAllWindows()

```

Design An AI Model for Translating Hand Gesture Recognition to Speech and Text

Gokul Kumar S, Jeberson Retna Raj, Jacob Raja T.
Department of Information Technology.
Sathyabama University.
Chennai, India.
jebersonretnarajr@gmail.com.

Abstract - It is estimated that there are millions of deaf-dumb people in the world communicating by sign language for their personal and professional life. It is very difficult to other people to understand the language. Therefore, to design a sign language recognition system is very meaningful and valuable for normal people to understand them. In this paper, a real-time system for sign language recognition is proposed to. As a first step, a sign language dataset is firstly created. Considering practical applications, a RGB camera is used to collect video stream, instead of a RGB-D camera. In order to improve the accuracy of recognize, we propose an optical flow method that uses 3D- CNN.

After the RGB video stream is collected, it is processed by dense optical flow, and then it is put into 3D- CNN to extract the feature vector. Real-time sign language recognition system with artificial interaction interface, motion detection module, hand and head detection module, etc, is designed for practical purposes. The proposed systems are demonstrated to perform excellently and are highly applicable.

Keywords - Sign language recognition, RGB, optical flow, 3D-CNN, motion detection, hand and head detection.

I. INTRODUCTION

In this world of continuously evolving technology, there is a need to interact with the technologies in a more improvised manner. As most of us have watched many science fiction movies, we must have noticed one of the most efficient ways of communicating with the technology is by using different fingers on our hands. The means of communicating with others with the help of nonverbal messages through the extensive use of finger motions is known as Hand Gestures. Such hand gestures come to us naturally and play a vital role in our life. By using it, one can express well and

can maintain the flow of a conversation. Others perceive us more clearly when we use hand gestures while communicating with them. Even blind people use them automatically, like a reflex action.

Hand Gesture recognition (HGR) is considered one of the main areas contributing a lot to the future of technology. One of the most advanced topics is HGR to speech conversion, which will help those with speech disabilities. In our work, we shall be using Flex sensors to classify different gestures. A Flex sensor, also known as a bend sensor is a low cost sensor which basically acts as a variable resistor. On bending the surface of the sensor, the resistance is varied which gives necessary readings. When the sensor is bent, the resistance starts increasing; at 90 degrees, the resistance is maximum.

II. LITERATURE SURVEY

Sergio Escalera et.al implemented the Flex sensors and Arduino UNO to recognize hand gestures and the data acquired from the sensors is analysed using traditional machine-learning algorithms. Compared to these traditional learning models, we developed an adversarial learning approach that performs more accurately. [1]

Koller et.al presented a real-time system based on 3D CNN method for recognizing the sign language. A sign language dataset initially created. For practical applications, RGB cameras are used instead of RGB-D cameras to collect video streams. In order to improve the accuracy of recognition, a 3D-CNN method combined with optical flow processing. [2]

In [3], the article can present an Arduino glove-based device is used for the conversion of sign language (ASL) to speech. The sign language glove consists of a simple hand glove fitted with flex sensors, which are used to monitor the amount of bend on the fingers. The analog signals from the sensors are digitally converted and compared with

the stored value for recognition of signs.

In [4], this paper can explain research provides an arduino-based framework used to convert sign language into sound language for bridging the communication gap between deaf mutes and the wider community. The system consists of Flex sensors and micro-controller's boards used to recognize hand/figure motion and translate it to sound.

Priyanka Lokhande et.al, presented a gesture recognition system for translating gesture into speech for other community people. The system employs with smart glove act as a interface for people with hearing or speech disabilities to communicate. The Gesture recognition can be done in two ways, image processing and sensors. The glove uses Flex sensor, Accelerometers, and Touch sensor to detect different gesture in sign language. Using Google Text-Speech, the data will be converted into speech. [5]

Gaolin Fang et.al, presented the device consists of flex sensors for detecting finger movements and an accelerometer for detecting arm movements. The system designed for identifying the Indian sign language and translate the sign into text and speech.[6]

Houqiang Li et.al, Implements a low cost control device called Hand Gesture or by using touch sensor for the patients to control the wheelchair and they can survey themselves for their basic needs and they need not to depend on others to guide those patients. The system was developed to indicate the directions for the patients to move on the flat surface and also measure the patient heart beat pulse.[7]

In [8], this work consists of a glove equipped with sensors which senses different sign language gesture, these senses data are fed to Arduino and transfer data to android phone via Bluetooth module, a common android phone is used in this work for sign to voice translation and voice to sign language translation.

In [9], this article can explain mainly on the hardware design of the device. Based on preliminary experimental results on nine (9) simple sign languages, the device demonstrated an average value of 0.74 s to convert a sign language into speech and text, which demonstrate the usefulness of the proposed device. This project describes the development of a sign language translator that converts sign language into speech and text by using wearable device. The glove-based device is able to read the movements of a single arm and five (5) fingers.

In [10], the system is working on a glove-based device which will be used for conversion of sign

language (ASL) to speech. The basic system consists of two parts; sign language recognition and conversion to text and further to speech. The sign language glove consists of a simple hand glove fitted with flex sensors which is being used for the monitoring the amount of bend on the fingers. Flex means bend, this is the sensors that change the resistance depending on the amount of bend on the sensor.

In [11], this paper can propose an idea which will assist in removing or at least reducing this gap between the speech impaired and normal people. The research going on this area mostly focuses on image processing approaches. Our prototype involves Arduino Uno as a microcontroller which is interfaced with flex sensors and accelerometer, gyroscopic sensor for reading the hand gestures. Furthermore, to perform better execution, we have incorporated an algorithm for better interpretation of data and therefore producing more accurate result. Thereafter, we use python to interface Arduino Uno with a microprocessor and finally converting into speech.

In [12], the goal of this study is to create a real-time system for hand gesture recognition that recognises hand gestures, hand features such as peak calculation and angle calculation, and then converts gesture images into voice and vice versa. We use a simple night vision webcam with a resolution of 20 megapixels to implement this system.

In [13], this paper can present a system prototype that is able to automatically recognize sign language to help normal people to communicate more effectively with the diminished people. The Sign to Voice system prototype, S2V, was developed using Feed Forward Neural Network for two-sequence signs detection. Mute people can't speak and normal people don't know the sign language which is used for inter communication between mute people.

In [14], this paper can propose a machine learning model which successfully and efficiently classifies hand gestures into spellings and sentences. Training and testing data spaces are kept different to yield better result confidence. CNN algorithms are applied aggressively to improve the conversion accuracy. A very important category of computational intelligence with respect to social applicability is the ability to transform any linguistic sign gesture into corresponding language text. This is formally known as Sign Language translation (SLT).

In [15], Hand gestures play an important role in human-computer interactions. The term "deep learning" refers to a type of machine learning that uses convolutional neural networks to make recognition easier. Convolutional Neural Networks are multilayer process networks that include an input layer, a convolution layer, a max pooling layer, a fully connected layer, and an output layer. CNN can provide more accurate results when

compared to other algorithms.

In [16], This paper can present, the system's goal is convert speech to American Sign Language using an RNN (Recurrent Neural Network) with LSTM (Recurrent Neural Network - Long Short-Term Memory) neural network trained by a Connectionist Temporal Classification (CTC) neural network is a gesture-based communication to text/speech model that enables a two-way correspondence without the use of an interpreter.

Niranjan panigrahi et.al, presented used a Computer Vision, Machine Learning, and Deep Learning methodologies to adapt ISL recognition in this work. To meet this requirement, the captured image is subjected to a series of pre-processing steps that include various Computer Vision techniques such as gray-scale conversion and thresholding using the OTSU algorithm. To train the system, Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), and pre-trained models, VGG-19 and Inception-V3, employ the Transfer Learning mechanism. [17]

In [18], this paper presents a different methods are glove based system, data input device and handicap touch screen methods are used. For text to speech conversion, microcontroller, sensors and measuring devices are used. These methods are expensive and difficult to implement.

In [19], this is a first component, Hand Gesture Detector, uses pre-trained models to detect hand gestures. The Image Classifier component categories and translates the hand signs that are detected. For identified hand signs, the Text and VoiceGenerator component generates text or audio formatted output. Finally, Text to Sign Converter converts entered English text into animated images based on sign language. Easy talk can detect, translate, and produce relevant outputs with greater accuracy by employing these techniques. This can lead to more effective and efficient communication between the community of people with disabilities and the community of people without disabilities.

In [20], this article can explain mainly the selection of user-accessible modes, a gesture recognition model based on the SVM algorithm is used. In addition, the system modes send and receive text data from an Android application via a cloud network. The received and sent data will be converted into Morse code pattern and voice form vibrations, respectively.

In [21], this is based on an optimised multilayer artificial neural network approach, the goal of this document is to develop and demonstrate the operation of a prototype capable of recognising the alphabet of Peruvian Sign Language and translating it into the Spanish language alphabet. It is concluded that this prototype is capable of translating Peruvian Sign Language with a high percentage of accuracy, as well as being viable and cost-effective, so testing the effectiveness of this

prototype in multiple people with speech disabilities is recommended.

In [22] this paper presents a employs with two sensors: a Microsoft Kinect and a Leap Motion Controller. In this paper, real-time hand gesture letters are recognised with the Kinect SDK Version 2 and LMC using a supervised learning method and natural user interface libraries.

In [23], this paper provides a portable and smart solution to meet the demand of turning disabled people's gestures into a text and speech synthesis format that is easily understood by others. This Smart Glove also has the ability to manage home appliances and monitor your health. The user will have an application installed that will assist them in switching between several modes of operation of our Smart gloves make life easier and more independent for people with special needs.

In [24], The Gesture detection mistreatment video and image process is employed for sanctioning the communication between the deaf, dumb and traditional folks. All the obtainable systems aren't moveable and not reasonable to poor folks. this paper introduce, the history of communication technologies that have given higher access to the planet for those sensory disabilities that may be are high communication technologies that improve the communication method of deaf and dumb like persons and create them advanced to speak with the opposite traditional persons.

In [25], the motive is to convert the human sign language to Voice by the help of some device which could be achieved with the help of flux and arduino combination. There are systems available for sign to speech conversion however none of them provide good user interface. For example if a person who has a disability to speak can stand perform the system and the system converts the human gestures as speech and plays it loud so that the person could actually communicate to a mass crowd gathering. Also the system could bring high efficiency for users to improve communication.

III. PROPOSED ARCHITECTURE

Sign languages are complex systems that include background noise. Convolutional neural networks should be applied to the picture for the best results in boosting categorization accuracy and for practical uses.

Most object identification issues involve the training of models using data from the picture data collection and a bounding box projection. Each image's enclosing frame must be marked, which is costly. Additionally, we put forth an area of interest prediction built on epidermis segmentation. We trim the picture and send it to the algorithm for prediction from the split limited area.

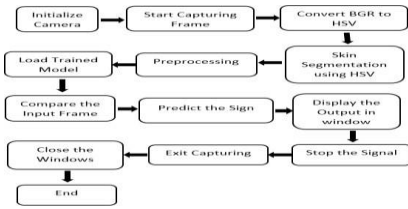


Fig. 1. Architecture of proposed model

We connected this potent machine learning tool with the hardware (Embedded Systems) in our project program, which was written using Python, to make it more effective, efficient, and to deliver real-time solutions. Facial recognition plays a significant part in our project. The result is transmitted via serial communication, which is connected to the microcontroller through a computer. Once that procedure is finished, the user has the choice to authenticate or not. If authentication is successful, the door will open; otherwise, it will stay closed.

A. Convolutional neural networks (CNN)

Deep Learning is showing to be an incredibly potent instrument because it can handle huge quantities of data. Pattern recognition has benefited from the use of hidden layers more than traditional methods. One of the most well-known profound brain networks is the convolutional brain organization. Convolutional brain networks are formed of many artificial neuronal layers. They are algebraic operations that combine inputs to execute tasks and produce an activation value. A shoddy replica of their real equivalents, artificial neurons. When you enter a picture into the convolution network, several activation functions are generated for each layer.

From the first layer, a fundamental characteristic, like a horizontal or vertical border, is removed. The following layer finds more intricate characteristics, like angles and combo edges, using this output. As we delve further into the network, we are able to recognise more intricate characteristics like items, people, and so forth. Based on the activation map produced by the final convolution layer, a categorization layer assesses the probability that a picture corresponds to a class. (Values from 0 to 1). For instance, a ConvoNet can recognise the activation map of the final convolution layer to identify whether an incoming picture includes a cat, canine, or equine.

B. Convolution layer:

A CNN's convolutional layer serves as its primary structural component. Learning the training procedure. (Or

C. Pooling layer:

The geographic area of the Convolved Feature is diminished by the Pooling layer. By decreasing the measurements, the amount of processing capacity needed to analyse the data will be reduced. The two pooling varieties are normal pooling and greatest pooling. I've quite recently used Max Pooling once, so far I haven't encountered any troubles.

We employ a method known as max pooling to choose the pixel with the most noteworthy worth from a piece of the picture covered by the part. Max Pooling likewise functions as a commotion suppressant. Furthermore, dimensionality decrease and de-noising are done, and all commotion enactments are killed. Normal Pooling, then again, gives the normal of each of the numbers from the district of the picture that the Portion takes care of. As a clamor dropping method, Normal Pooling just lessens aspects. Therefore, we can infer that Maximum Pooling performs essentially better compared to Average Pooling.

D. Second Convolution layer:

The result of the first convolutional layer in a convolutional brain organization (CNN) is exposed to a progression of channels in the second convolutional layer. The result of the preceding layer is convolved with each filter's collection of weights to create a set of activation maps.

The second convolutional layer's purpose is to extract more complex and abstract features from the input image. The second layer's filters learn to detect more sophisticated patterns and shapes than the first layer's, building on the features discovered in the previous layer.

Overall, the second convolutional layer plays a crucial role in the CNN architecture, enhancing the network's capacity to carry out tasks like picture categorization, object recognition, and segmentation by teaching it increasingly complicated models of the incoming data.

E. Flattening:

There is no flattening layer in the Convolutional Neural Network (CNN) algorithm. Flattening is an operation that is performed on the result of the convolutional and pooling layers, for the most part before the completely associated layers.

The flattening operation converts the 3D tensor result of the convolutional and pooling layers into a 1D vector that can be handled by the fully connected layers. This is accomplished by rearranging the tensor into a long vector by simply concatenating all of the tensor's values into a single

dimension. For example, if the output tensor has dimensions of (batch size, height, width, depth), the flattened tensor will have the shape of (batch size, height x width x depth).

Overall, flattening is an important operation in the CNN algorithm that enables the extraction of important features from input images and their transformation into a format suitable for classification or regression tasks.

F. Fully Connected Layer:

A Convolutional Neural Network (CNN) layer is known as a fully linked (FC) layer on the off chance that every neuron in the layer is associated with every neuron in the layer above it. In contrast to the convolutional and pooling layers, which operate on spatial features, the FC layer operates on the flattened output of the preceding layers.

The purpose of the FC layer is to use the extracted features from the previous layers to make a prediction about the input image. The FC layer is typically placed at the end of the CNN architecture, after significant characteristics from the raw data have been recovered by the convolutional and pooling layers. The output of the FC layer is a probability distribution over the possible classes, which is then used to make a prediction.

Overall, the Fully Connected layer plays a critical role in the CNN algorithm, allowing the network to use the extracted features from the previous layers to make a prediction about the input data.

IV. ARCHITECTURE OF ADVERSARIAL LEARNING MODEL

For picture identification tasks, a form of neural network called a convolutional neural network (CNN) is frequently used.

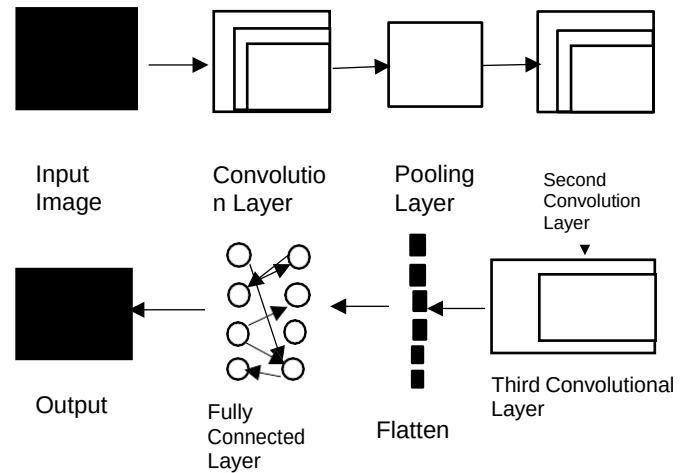
Data Preparation: Preparing the data is the first stage in using a CNN for motion detection. This typically involves collecting a large number of images of the different gestures that the system will recognize. These images may need to be preprocessed to adjust the size, brightness, and contrast, and to remove any irrelevant features.

Training the Model: After the data has been collected, the CNN model must be trained. In order to do this, the data must be divided into a preparation set and an affirmation bunch. While the approval set is utilized to survey the model's exhibition, the preparation set is utilized to instruct the CNN how to recognise various motions.

Convolutional Layers: The core of the CNN algorithm is the convolutional layers. These layers use filters to identify patterns in the input data. In the case of gesture recognition, the filters might look for specific shapes or patterns that are associated with different gestures.

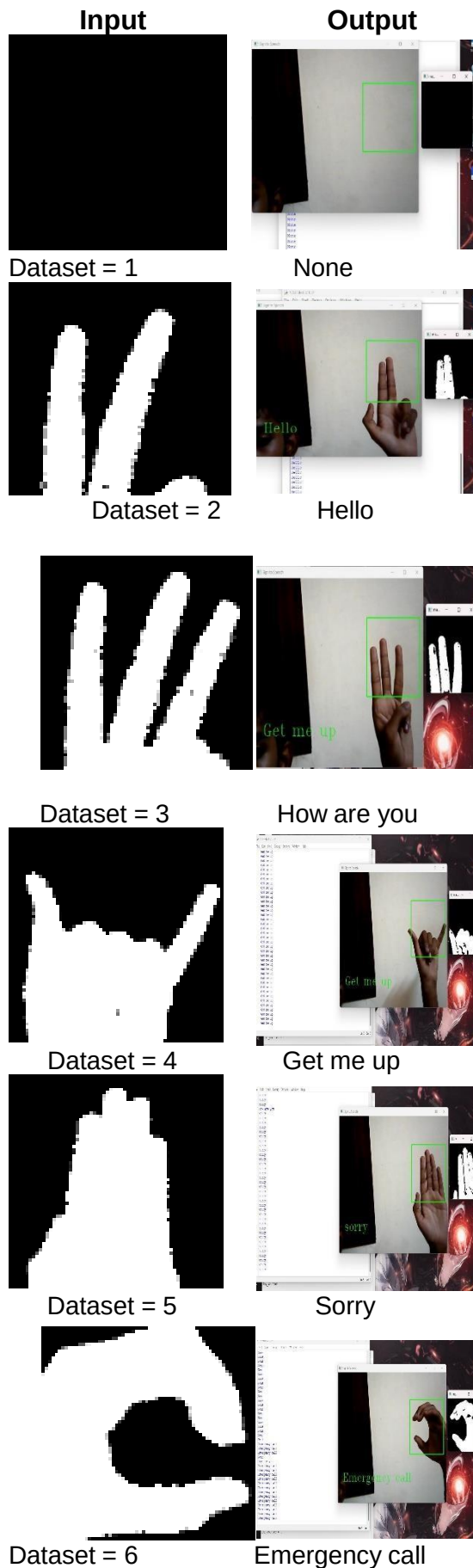
Pooling Layers: Typically, a pooling layer is inserted after each convolutional layer. The pooling layer's goal is to decrease the output's complexity while preserving the key characteristics from the convolutional layer.

Fully Connected Layers: Once the data has been processed through the convolutional and pooling layers, it is passed through one or more fully connected layers. These layers use the features extracted from the previous layers to make a final decision about which gesture is being performed.



V. EXPERIMENTAL RESULT

Using Keras, a Python tool frequently used for creating deep learning models, the complete experiment's code was created. Scikit-learn, pandas, and NumPy were some of the additional tools used. The trial called for using a computer system with a specialised Graphical Processing Unit (GPU), so an AMD Ryzen 5 3550H CPU and Nvidia GeForce GTX 1050 graphics card were used on the machine. It was discovered that the adversarial learning beat traditional classifications. All machine learning methods were applied to the information we got from our original research in order to perform the evaluations. In order to have a perfect analysis of all the versions being compared, we have made every effort to maintain as many of the required circumstances as comparable as feasible. The success index of the ANN used prior to the use of Adversarial Learning on our dataset is displayed in the chart below



VI. OBSERVATION OF TESTING AFTER APPLICATION OF ADVERSARIAL LEARNING

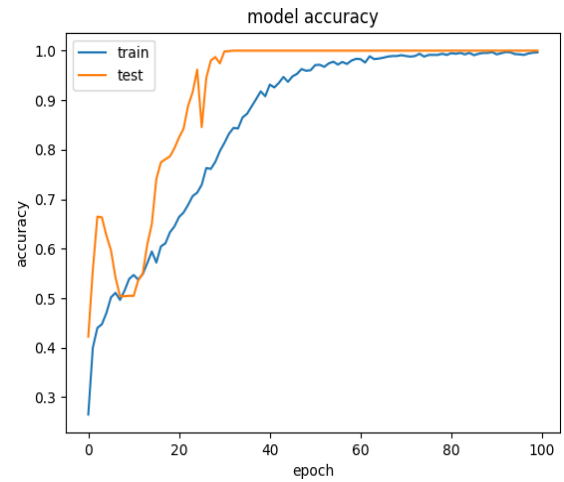


Fig.2 Training Accuracy

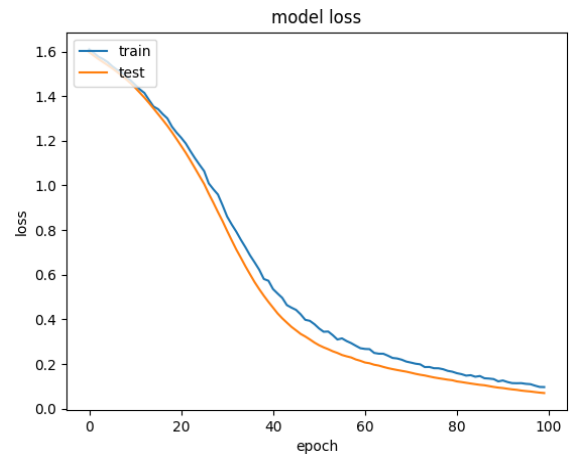


Fig.3 Validation Loss

PERFORMANCE OF THE PROPOSED METHOD

Algorithms	Training Accuracy	Validation Loss
CNN Algorithm	0.99	0.43

VI. CONCLUSION

Healthcare has never before experienced anything like the effect that hand gesture recognition has had. This inspires us to create a categorization model that will serve social requirements for the same. In conclusion, we hope to enhance the model's output by enlarging the sample size of the dataset that was taken. Additionally, by tweaking the current model and its associated restrictions, we try to make the method work better. We truly believe that this will help people with communication impairments.

VII. REFERENCE

- [1] P. Surekha, N. Vitta, P. Duggirala and V. S. S. Ambadipudi, "Hand Signal Acknowledgment and voice, text change utilizing," 2022 Second Global Gathering on Man-made brainpower and Savvy Energy (ICAIS), Coimbatore, India, 2022, pp. 167-171, doi: 10.1109/ICAIS53314.2022.9743064.
- [2] Gao, Wen and Tooth, Gaolin and Zhao, Debin and Chen, Yiqiang. (2004). A Chinese communication via gestures acknowledgment framework in light of SOFM/SRN/Gee. Design Acknowledgment. 37. 2389-2402. 10.1016/j.patcog.2004.04.008.
- [3] Huang, Jie and Zhou, Wengang and Li, Houqiang and Li, Weiping. (2015). Gesture based communication Acknowledgment utilizing 3D convolutional brain organizations. 1-6. 10.1109/ICME.2015.7177428.
- [4] Zach, Christopher and Scar, Thomas and Bischof, Horst. (2007). A Duality Based Approach for Realtime television L1 Optical Stream. Design Acknowledgment. 4713. 214-223. 10.1007/978-3-540-74936-3_22.
- [5] M R, Pooja and M, Meghana and Bhaskar, Harshith and Hulatti, Anusha and Koppalkar, Praful and J, Bopanna. (2022). Communication via gestures Acknowledgment Framework. Indian Diary of Computer programming and Venture The executives. 1. 1-3. 10.35940/ijsepm.C9011.011322.
- [6] Koller, Oscar and Forster, Jens and Ney, Hermann. (2015). Consistent gesture based communication acknowledgment: Towards huge jargon measurable acknowledgment frameworks dealing with different underwriters. PC Vision and Picture Getting it. 141. 108-125. 10.1016/j.cviu.2015.09.013.
- [7] Fronteddu, Graziano and Porcu, Simone and Floris, Alessandro and Atzori, Luigi. (2022). A Unique Hand Signal Acknowledgment Dataset for Human-PC Points of interaction. PC Organizations. 205. 108781. 10.1016/j.comnet.2022.108781.
- [8] Georgi, Marcus, Christoph Amma, and Tanja Schultz". Perceiving Hand and Finger Signals with IMU based Movement and EMG based Muscle Action Detecting." In Biosignals, pp. 99-108. 2015.
- [9] artificial brain network in light of electromyography (EMG) sensor." In 2015 Worldwide Meeting on Mechanization, Mental Science, Optics, Miniature Electro-Mechanical Framework, and Data Innovation (ICACOMIT), pp. 12-17. IEEE, 2015.
- [10] Almeida, FG Guimarães, JA Ramírez. Feature extraction in Brazilian Sign Language Recognition based on phonological structure and using RGB-D sensors[J]. Expert Systems with Applications, 2014, 41(16):7259-7271.
- [11] Alosail, H. Aldolah, L. Alabdulwahab, A. Bashar and M. Khan, "Smart Glove for Bi-lingual Sign Language Recognition using Machine Learning," 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 2023, pp. 409-415, doi: 10.1109/IDCIoT56793.2023.10053470.
- [12] S. S. Shinde, R. M. Autee and V. K. Bhosale, "Real time two way communication approach for hearing impaired and dumb person based on image processing," 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Chennai, India, 2016, pp. 1-5, doi: 10.1109/ICCIC.2016.7919572.
- [13] "Prototype for Peruvian Sign Language translation based on an artificial neural network approach," 2020 IEEE XXVII International Conference on Electronics, Electrical Engineering and Computing (INTERCON), Lima, Peru, 2020, pp. 1-4, doi: 10.1109/INTERCON50315.2020.9220257.
- [14] Y. Manoharan, S. Saxena and D. R., "A Vision-based Smart Human Computer Interaction System for Hand-gestures Recognition," 2022 1st International Conference on Computational Science and Technology (ICCTST), CHENNAI, India, 2022, pp. 321-324, doi: 10.1109/ICCTST55948.2022.10040464.
- [15] S. Pariselvam, D. N., D. S. and S. B., "An Interaction System Using Speech and Gesture Based on CNN," 2020 International Conference on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, 2020, pp. 1-5, doi: 10.1109/ICSCAN49426.2020.9262343.
- [16] U. Om Kumar, K. P. K. Devan, P. Renukadevi, V. Balaji, A. Srinivas and R. Krithiga, "Real Time Detection and Conversion of Gestures to Text and Speech to Sign System," 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2022, pp. 73-78, doi: 10.1109/ICESC54411.2022.9885562.
- [17] N. Panigrahi, "Artificial Intelligence based Indian Sign Language Recognition with Accelerated Performance under HPC Environment," 2022 OITS International Conference on Information Technology (OCIT), Bhubaneswar, India, 2022, pp. 228-232, doi: 10.1109/OCIT56763.2022.00051.
- [18] Sunitha K. A, Anitha Saraswathi.P, Aarthi.M, Jayapriya. K, Lingam Sunny, "Deaf Mute Communication Interpreter- A Review", International Journal of Applied Engineering Research, Volume 11, pp 290-296, 2016.
- [19] Manoj Kumar, K. Bavanraj, S. Thavananthan, G. M. A. S. Bastiansz, S. M. B. Harshanath and J. Alosious, "EasyTalk: A Translator for Sri Lankan Sign Language using Machine Learning and Artificial Intelligence," 2020 2nd International Conference on Advancements in Computing (ICAC), Malabe, Sri Lanka, 2020, pp. 506-511, doi: 10.1109/ICAC51239.2020.9357154.
- [20] B. J. Kranthi, G. Suhas, K. B. Varma and G. P. Reddy, "A two-way communication system with Morse code medium for people with multiple disabilities," 2020 IEEE 7th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Prayagraj, India, 2020, pp. 1-6, doi: 10.1109/UPCON50219.2020.9376479.
- [21] J. E. Mejía Gamarra, M. Alonso Salazar Cubas, J. D. Sosa Silupú and C. Enrique Córdova Chirinos, "Prototype for Peruvian Sign Language translation based on an artificial neural network approach," 2020 IEEE XXVII International Conference on Electronics, Electrical Engineering and Computing (INTERCON), Lima, Peru, 2020, pp. 1-4, doi: 10.1109/INTERCON50315.2020.9220257.
- [22] Almasre, Miada A., and Hana Al-Nuaim. "A Real-Time Letter Recognition Model for Arabic Sign Language Using Kinect and Leap Motion Controller v2." *International Journal of Advanced Engineering, Management and Science*, vol. 2, no. 5, May. 2016.
- [23] K.V.Fale, Akshay Phalke, Pratik Choudhari and Pradeep Jadhav, "SMART GLOVE: Gesture Vocalizer for Deaf and Dumb People", International Journal of Innovative research in Computer and Communication Engineering, Vol.5, Issue 5, pp. 6800-6806, April 2016.
- [24] S. Vigneshwaran, M. Shifa Fathima, V. Vijay Sagar and R. Sree Arshika, "Hand Gesture Recognition and Voice Conversion System for Dumb People," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 2019, pp. 762-765, doi: 10.1109/ICACCS.2019.8728538.

- [25] Choudhary, Dhirendra Kumar and Singh, Rameet and Kamthania, Deepali, Sign Language Recognition System (April 22, 2021). Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2021