REAL TIME RECOGNITION OF SIGN LANGUAGE

A PROJECT REPORT

Submitted by

SHARON SAGHANA - 312318205140
SHEKHINA MARY J - 312318205142

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

St. JOSEPH’S COLLEGE OF ENGINEERING

(An Autonomous Institution)

St. Joseph’s Group of Institutions

Jeppiaar Educational Trust

OMR, Chennai 600 119

ANNA UNIVERSITY: CHENNAI

May-2022

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **“REAL TIME RECOGNITION OF SIGN LANGUAGE**” is the bonafide work of **SHARON SAGHANA (312318205140)** and **SHEKHINA MARY J (312318205057)** who carried out the project under my supervision.

SIGNATURE
Supervisor,
Mr. Raja Mohamed N, M.E.,

Assistant Professor,
Department of IT,
St.Joseph’s College of Engineering,
OMR, Chennai- 600119.

SIGNATURE
Head of the department,
Dr. V Muthulakshmi, M.E.,Ph.D,

Associate Professor,
Department of IT,
St.Joseph’s College of Engineering,
OMR, Chennai- 600119.

CERTIFICATE OF EVALUATION

COLLEGE NAME **:** St. Joseph's College of Engineering, Chennai-600119.

BRANCH **: B.TECH., IT** (Information Technology)

SEMESTER VIII

SL. NO	NAME OF THE STUDENT	TITLE OF THE PROJECT	NAME OF THE SUPERVISOR WITH DESIGNATION
1 2	SHARON SAGHANA (312318205140) SHEKHINA MARY J (312318205142)	REAL TIME RECOGNITION OF SIGN LANGUAGE	Mr. RAJA MOHAMED N, M.E., ASSISTANT PROFESSOR

The report of the project work submitted by the above students in partial fulfillment for the award of Bachelor of Technology Degree in Information Technology of Anna University was confirmed to be report of the work done by the above students and then evaluated.

Submitted to Project and Viva Examination held on________________.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

At the outset we would like to express our sincere gratitude to the beloved **Chairman, Dr.Babu Manoharan, M.A.,M.B.A.,Ph.D.,** for his constant guidance and support.

We would like to express our heartfelt thanks to our respected **Managing Director Mrs. S. Jessie Priya, M.Com.,** for her kind encouragement and blessings.

We wish to express our sincere thanks to our **Executive Director Mr. B. Shashi Sekar, M.Sc.,** for providing ample facilities in the institution.

We express our deepest gratitude and thanks to our beloved **Principal Dr.Vaddi Seshagiri Rao, M.E., M.B.A., Ph.D., F.I.E.,** for his inspirational ideas during the course of the project.

We wish to express our sincere thanks and gratitude to **Dr.V. Muthulakshmi, M.E.,Ph.D.,** Head of the Department , Department of Information Technology, St. Joseph's College of Engineering for her guidance and assistance in solving the various intricacies involved in the project.

It is with deep sense of gratitude that we acknowledge our indebtness to our supervisor **Mr. Raja Mohamed N, M.E.,** for his expert guidance and connoisseur suggestion.

Finally we thank our department staff members who helped us in the successful completion of this project.

ABSTRACT

There is an absence of communication with deaf people in our society. To overcome this barrier the introduction of Sign Language (SL) took place. To convey meaning to normal people, sign language makes use of patterns that are visually transmitted sign patterns. Sign language is also useful for people suffering with Autism Spectrum Disorder (ASD). Normal people cannot understand the signs used by deaf, as they do not know the meaning of a particular sign. The system proposed here aims at solving this problem. This system uses a camera, which captures various gestures of the hand. Then, processing of the image takes place by using various algorithms. First, pre-processing of the image takes place. Then, determination of edges occurs by using an edge detection algorithm. Finally, a template-matching algorithm identifies the sign and display the text. As the output is text, one can easily interpret the meaning of a particular sign. This also curtails the difficulty to communicate with the deaf. The implementation of the system is by using OpenCV-Python along with the usage of various libraries to implement the same.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

S.NO	ABBREVIATION	DEFINITION
1	ML	Machine Learning Algorithm
2	CNN	Convolutional Neural Network
3	SLR	Sign Language Recognition
4	SVM	Support Vector Machine
5	PCA	Principle Component Analysis
6	HMM	Hidden Markov Model

CHAPTER-1

1.1 INTRODUCTION

In recent years, technologies like gesture recognition, facial recognition have received huge importance under the branch of sign language. Gestures are various movements used in the process of communication. Either hand or body makes gestures. Sign language makes use of gestures, which usually make use of visually transmitted patterns. All over the world, the people suffering from hearing problems have a count of 4,94,93,50,000 approximately. Some of the existing sign language conversion systems consider hand orientation, hand shape and hand movement. In sign language, every sign has a meaning assigned to it, so that it becomes easy to understand and interpret by the people. The people, based on their language and the place in which they live, develop discrete and non-identical sign languages. There is no sign language accepted universally. People use various sign languages across the world.

Indian Sign Language presents various hand movements by using both right and left hands. The proposed work focuses on hand shape and orientation and works on American Sign Language. When using ASL, only one hand is used. Therefore, it becomes easy for implementing the system. ASL does not depend on any of the spoken languages and it has its own path of development.

The process in brief includes acquiring images using a camera. Then performing pre-processing steps on the image, that is, convert the acquired image, which is in RGB model to gray scale image. Later, track the edges by using canny edge detection algorithm. Finally, detecting the pattern using template-matching algorithm, this outputs the result as text. This system bridges the imbalances between deaf people and normal people without any requirement of an intermediate translator. It achieves the objective of conversion of gestures to text. Before going to the methodology, have a glance at literature survey.

1.2 SIGN LANGUAGE

It is a language that includes gestures made with the hands and other body parts, including facial expressions and postures of the body.It used primarily by people who are deaf and dumb. There are many different sign languages as, British, Indian and American sign languages. British sign language (BSL) is not easily intelligible to users of American sign Language (ASL) and vice versa .

A functioning signing recognition system could provide a chance for the inattentive communicate with non-signing people without the necessity for an interpreter. It might be wont to generate speech or text making the deaf more independent. Unfortunately there has not been any system with these capabilities thus far. during this project our aim is to develop a system which may classify signing accurately.

American Sign Language (ASL) is a complete, natural language that has the same linguistic properties as spoken languages, with grammar that differs from English. ASL is expressed by movements of the hands and face. It is the primary language of many North Americans who are deaf and hard of hearing, and is used by many hearing people as well.

1.3 OVERVIEW OF THE PROJECT

In many ways, sign languages are like spoken languages: they are natural languages that arise spontaneously wherever there is a community of communicators; they effectively fulfill all the social and mental functions of spoken languages; and they’re acquired without instruction by children, given normal exposure and interaction. These characteristics have led many linguists to expect sign languages to be similar to spoken languages in significant ways. But sign languages are different too: as manualvisual languages, sign languages exploit a completely

different physical medium from the vocal-auditory system of spoken language.These two dramatically different physical modalities are also likely to have an effect on the structure of the languages through which they are transmitted.

It is of special interest, then, to compare natural languages in the two modalities. Where the two systems converge, universal linguistic properties are revealed. Where they diverge, the physical medium of transmission is implicated, and its contribution to the form of language in both modalities illuminated. Neither can be seen quite so clearly if linguists restrict their study to spoken language alone (or to sign language alone). For this and other related reasons, it is often remarked that sign languages provide us with a natural laboratory for studying the basic characteristics of all human language.

Once the existence of natural language in a second modality is acknowledged, questions like the following arise: How are such languages born? Are the central linguistic properties of sign languages parallel to those of spoken languages? Is sign language acquired by children in the same stages and time frame in which hearing children acquire spoken language? Are the same areas of the brain responsible for language in both modalities? What role does modality play in structuring language? In other words, within the architecture of human cognition, do we find the structure of one language ‘faculty’ or two? While there is no conclusive answer to this deceptively simple question, an impressive body of research has greatly expanded our understanding of the issues underlying it.

1.4 SCOPE OF PROJECT

Computer recognition of sign language is an important research problem for enabling communication with hearing impaired people. This project introduces an efficient and algorithm for identification of the number of fingers opened in a gesture representing an alphabet of the Binary Sign Language. The system does not require the hand to be perfectly aligned to the camera. The project uses image processing system to identify, especially English alphabetic sign language used by the deaf people to communicate. The basic objective of this project is to develop a computer based intelligent system that will enable dumb people significantly to communicate with all other people using their natural hand gestures. The idea consisted of designing and building up an intelligent system using image processing, machine learning and artificial intelligence concepts to take visual inputs of sign language's hand gestures and generate easily recognizable form of outputs. We can develop a model for ISL word and sentence level recognition. This will require a system that can detect changes with respect to the temporal space. We can also develop a complete product that will help the speech and hearing impaired.

CHAPTER 2

LITERATURE SURVEY

[1] Neha Poddar, Shrushti Rao, Shruti Sawant, Vrushali Somavanshi, Prof. Sumita Chandak "Study of Sign Language Translation using Gesture Recognition", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, Issue 2, 2015.

The mute/deaf individuals have a communication problem dealing with other people. It is hard for such individuals to express what they want to say since sign language is not understandable by everyone. This paper is to develop a Data Acquisition and Control (DAC) system that translates the sign language into text that can be read by anyone. This system is called Sign Language Translator and Gesture Recognition. We developed a smart glove that captures the gesture of the hand and interprets these gestures into readable text. This text can be sent wirelessly to a smart phone or shown in an embedded LCD display. It is evident from the experimental results that gestures can be captured by set of inexpensive sensors, which measure the positions and the orientation of the fingers. The current version of the system is able to interpret 20 out of 26 letters with a recognition accuracy of 96%.

[2] Sawant Pramada, Deshpande Saylee, Nale Pranita, Nerkar Samiksha, Mrs.Archana S, Vaidya "Intelligent Sign Language Recognition Using Image Processing", IOSR Journal of Engineering, Vol. 3, Issue 2, 2013.

Computer recognition of sign language is an important research problem for enabling communication with hearing impaired people. This project introduces an efficient and fast algorithm for identification of the number of fingers opened in a gesture representing an alphabet of the Binary Sign Language. The system does not require the hand to be perfectly aligned to the camera. The project uses image processing system to identify, especially English alphabetic sign language used by the deaf people to communicate. The basic objective of this project is to develop a computer based intelligent system that will enable dumb people significantly to communicate with all other people using their natural hand gestures. The idea consisted of designing and building up an intelligent system using image processing, machine learning and artificial intelligence concepts to take visual inputs of sign language's hand gestures and generate easily recognizable form of outputs. Hence the objective of this project is to develop an intelligent system which can act as a translator between the sign language and the spoken language dynamically and can make the communication between people with hearing impairment and normal people both effective and efficient. The system is we are implementing for Binary sign language but it can detect any sign language with prior image processing

[3] Prof. Supriya Pawar, Sainath Muluk, Sourabh Koli "Real Time Sign Language Recognition using Python", International Journal of Innovative Research in Computer and Communication Engineering ,Vol. 6, Issue 3, 2018.

Sign Language is the method of communication of deaf and dumb people all over the world. However, it has always been a difficulty in communication between a verbal impaired person and a normal person. Sign Language Recognition is a breakthrough for helping deaf-mute people to communicate with others. The commercialization of an economical and accurate recognition system is today's concern of researchers all over the world. Thus, sign language recognition systems based on Image processing and neural networks are preferred over gadget system as they are more accurate and easier to make. The aim of this paper is to build a user friendly and accurate sign language recognition system trained by neural network thereby generating text and speech of the input gesture. This paper also presents text to sign language generation model that enables a way to establish a two-way communication without the need of a translator.

[4] P.V.V.Kishore, "Segment, Track, Extract, Recognize and Convert Sign Language Videos to Voice/Text", International Journal of Advanced Computer Science and Applications, Vol. 3, Issue 6, 2012.

This paper summarizes various algorithms used to design a sign language recognition system. Sign language is the language used by deaf people to communicate among themselves and with normal people. We designed a real time sign language recognition system that can recognize gestures of sign language from videos under complex backgrounds. Segmenting and tracking of non-rigid hands and head of the signer in sign language videos is achieved by using active contour models. Active contour energy minimization is done using signers hand and head skin colour, texture, boundary and shape information. Classification of signs is done by an artificial neural network using error back propagation algorithm. Each sign in the video is converted into a voice and text command. The system has been implemented successfully for 351 signs of Indian Sign Language under different possible video environments. The recognition rates are calculated for different video environments.

[5] Kamal Preet Kour, Dr. Lini Mathew" Sign Language Recognition Using Image Processing", International Journals of Advanced Research in Computer Science and Software Engineering, Vol. 7, Issue 8, 2017.

One of the major drawback of our society is the barrier that is created between disabled or handicapped persons and the normal person. Communication is the only medium by which we can share our thoughts or convey the message but for a person with disability (deaf and dumb) faces difficulty in communication with normal person. For many deaf and dumb people , sign language is the basic means of communication. Sign language recognition (SLR) aims to interpret sign languages automatically by a computer in order to help the deaf communicate with hearing society conveniently. Our aim is to design a system to help the person who trained the hearing impaired to communicate with the rest of the world using sign language or hand gesture recognition techniques. In this system, feature detection and feature extraction of hand gesture is done with the help of SURF algorithm using image processing. All this work is done using MATLAB software. With the help of this algorithm, a person can easily trained a deaf and dumb.

[6] Zhi-hua Chen, Jung-Tae Kim, Jianning Liang, Jing Zhang and Yu-Bo Yuan "Real-Time Hand Gesture Recognition Using Finger Segmentation", The Scientific World Journal, Volume 2014, Article ID 267872, 9

Hand gesture recognition is very significant for human-computer interaction. In this work, we present a novel real-time method for hand gesture recognition. In our framework, the hand region is extracted from the background with the background subtraction method. Then, the palm and fingers are segmented so as to detect and recognize the fingers. Finally, a rule classifier is applied to predict the labels of hand gestures. The experiments on the data set of 1300 images show that our method performs well and is highly efficient. Moreover, our method shows better performance than a state-of-art method on another data set of hand gestures.

[7] Neelam K. Gilorkar, Manisha M. Ingle, "Real Time Detection And Recognition Of Indian And American Sign Language Using Sift", International Journal of Electronics and Communication Engineering & Technology, Vol. 5, Issue 5, 2014.

Normal Human being is able to see and interact with the surroundings. Every individual are not blessed with this criteria. Minority of the people tend to lose their vocal cord in an accident. Such type of patient is known as 'Aphasia Patient'. The communication mannerism of aphasia patient doesn't coincide with the normal people. This project is to provide support for aphasia people by recognising their gestures and converting into its corresponding audio language. India is been developing in to the digitalized technology, so the difficulty can be overcome by the newly systems. The learning of the gestures postures differs from each individual, which may also lead to a wrong understanding of the conversation. This difficulty has to be overcome, in order to make the patient to live in a betterment life. The mute language is changed into the speech by the digital signal recognition method. The glove includes 4 flex sensors that gain the pressure produced in the fingers, GSM module to transfer the message for a particular cellular, IR sensor to control the objects in the room, LCD to show off the signs as the text message and a speaker that increases the amplitude of the gesture language. Initially, Flux sensor working is performed and result is identified. The performance of the sensor is up to 85% .The working of flux sensor could be improved.

[8] T. Ayshee, S. Raka, Q. Hasib, Md. Hossian, R. Rahman, "Fuzzy Rule-Based Hand Gesture Recognition for Bangali Characters", in IEEE International Advanced Computing Conference, 2014.

Sign language recognition is one of the most growing fields of research today and it is the most natural way of communication for the people with hearing problems. A hand gesture recognition system can provide an opportunity for deaf persons to communicate with vocal people without the need of an interpreter or intermediate. The system is built for the automatic recognition of Marathi sign language. Providing teaching classes for the purpose of training the deaf sign user in Marathi. The system can train new user who is unaware of the sign language and the training will be provided through offline mode. In which user can learn sign language with the help of database containing predefined sign language alphabets as well as words. A large set of samples has been used in proposed system to recognize isolated words from the standard Marathi sign language which are taken using camera. The system contains forty-six Marathi sign language alphabets and around 500 words of sign language are taken. Considering all the sign language alphabets and words, the database contains 1000 different gesture images. The proposed system intend to recognize some very basic elements of sign language and to translate them to text and vice versa.

[9] B. Bauer,H. Hienz "Relevant features for video-based continuous sign language recognition", IEEE International Conference on Automatic Face and Gesture Recognition, 2002.

This paper describes the development of a video-based continuous sign language recognition system. The system is based on continuous density hidden Markov models (HMM) with one model for each sign. Feature vectors reflecting manual sign parameters serve as input for training and recognition. To reduce computational complexity during the recognition task beam search is employed. The system aims for an automatic signer-dependent recognition of sign language sentences, based on a lexicon of 97 signs of German sign language (GSL). A further colour video camera is used for image recording. Furthermore the influence of different features reflecting different manual sign parameters on the recognition results are examined. Results are given for varying sized vocabulary. The system achieves an accuracy of 91.7% based on a lexicon of 97 signs.

[10] Patil, Prajakta M., and Y. M. Patil, "Robust Skin Colour Detection and Tracking Algorithm", International Journal of Engineering Research and Technology. Vol.1, Issue8, 2012

Sign language is the basic medium of communication for the deaf and dumb people. It is a language which uses manual communication and body language to convey meaning. This can involve combining hand shapes, orientation and movement of hands. Communication may be the biggest challenge for the deaf and dumb in order to receive and convey information, ideas and feelings. Thus, in order to bridge the gap between them and the others, it becomes necessary to build a communicator and translator to translate American Sign Language to Indian and vice versa. In addition to this, the American and Indian sign language is also converted to text and back. During this translation, in order to ensure efficient skin detection and further processing of image, the paper focuses on obtaining appropriate results on Indian sign images based on background removal algorithms. This paper presents the comparison among various color spaces for skin detection based on background removal from hand sign images. The color space is a useful way to specify and conceptualize the color capabilities of a particular digital file or an image. The proposed techniques on colour spaces are executed on a dataset of 78 images. In order to analyze the results of the image based on various color spaces, this study of comparison among them is needed. It elaborates mainly on four color models : RGB, YCbCr, HSV, NTSC. This paper analyses the results of the above color spaces.

The preliminary steps in skin detection are the representation of image pixels in color spaces, suitable distribution of skin & non skin pixels, & after that skin color modeling. According to skin color's distribution characteristics on color space, skin color pixels can be detected quickly with skin color model. However it is difficult to detect skin color more accurately, because there exists many differences about skin color space distribution, which is affected by different race and different illumination The preliminary steps in skin detection are the representation of image pixels in color spaces, suitable distribution of skin & non skin pixels, & after that skin color modeling. According to skin color's distribution characteristics on color space, skin color pixels can be detected quickly with skin color model.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In sign language recognition, Hidden Markov Models (HMMs) have been popularly exploited. In each sign is modeled with one HMM. For both training and recognition, feature vectors must be extracted from each video frame and then inputted to the HMM. Cotton gloves are used with several color-markings in signers, palm and back of the hand to get both trajectory and hand shape features from video. Described a HMM based system which used one color camera to track unadorned hands in real time and interpret American Sign Language using HMMs with a lexicon of 40 words.

3.1.1 Disadvantages

- Main difficulty is modeling probability of assigning a tag to word can be very difficult if “words” are complex.
- It is not practical to represent multiple overlapping features and long term dependencies.
- Number of parameters to be evaluated is huge. So it needs a large data set for training.
- It requires huge amount of training in order to obtain better results.

3.2. PROPOSED SYSTEM:

They applied the developed models to recognize human actions and achieved superior performance in comparison to baseline methods. Applied 3D CNNs to large scale video classification. Their spatiotemporal networks demonstrate significant performance improvements compared to strong feature-based baselines in a dataset of 1 million YouTube videos from 487 categories. In this paper, we develop a 3D CNN to integrate multisource of visual data. Each type of data stream provides several adjacent frames as input. In the experiments, we show that our model outperforms the baseline method based on hand-crafted features.

3.2.1 Advantages

- It automatically detects the important features without any human supervision.
- They are great at handling image classification.
- They use the same knowledge across all image locations.

3.3 SYSTEM SPECIFICATION

HARDWARE CONFIGURATION

• Processor	-	I5
• Speed	-	3 GHz
• RAM	-	8 GB(min)
• Hard Disk	-	500 GB

- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - SVGA

SOFTWARE CONFIGURATION

- Operating System: Linux, Windows/7/10
- Server: Anaconda, Jupyter,pycharm
- Front End: tkinter |GUI toolkit
- Server side Script: Python , AIML

3. 4. LANGUAGE SPECIFICATION

3.4.1 TensorFlow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google. Features: TensorFlow provides stable Python (for version 3.7 across all platforms) and C APIs; and without API backwards compatibility guarantee: C++, Go, Java, JavaScript and Swift (early release).

Third-party packages are available for C#, Haskell Julia, MATLAB,R, Scala, Rust, OCaml, and Crystal."New language support should be built on top of the C API.

However, not all functionality is available in C yet." Some more functionality is provided by the Python API. Application: Among the applications for which TensorFlow is the foundation, are automated image-captioning software, suchas DeepDream.

3.4.2 Opencv

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision

- Egomotion estimation
- Facial Recognition System
- Gesture Recognition
- Mobile Robotics
- Motion Understanding
- Object Identification
- Support Vector Machine.
- Segmentation and Recognition
- Deep Neural Networks
- AForget.NET
- Artificial Neural Networks
- Random Forest

a computer vision library for the Common Language Runtime (.NET Framework and Mono). 8 ROS (Robot Operating System). OpenCV is used as the primary vision package in ROS. VXL, an alternative library written in C++. Integrating Vision Toolkit (IVT), a fast and easy-to-use C++ library with an optional interface to OpenCV. CVIPtools, a complete GUI-based computer-vision and image-processing software environment.

VXL, an alternative library written in C++. Integrating Vision Toolkit (IVT), a fast and easy-to-use C++ library with an optional interface to OpenCV. CVIPtools, a complete GUI-based computer-vision and image- processing software environment, with C function libraries, a COM-based DLL, along with two utility programs for algorithm development and batch processing. OpenNN, an open-source neural networks library written in C++. List of free and open source software packages

- OpenCV Functionality
- Image/Video I/O, processing
- Object/ Feature detection
- Geometry based monocular or stereo computer vision
- Computational Photography
- Machine Learning
- Clustering
- CUDA Accleration

3.4.3 Image-Processing

Image processing is a method to perform some operations on an image, in order to get an enhanced image and or to extract some useful information from it. If we talk about the basic definition of image processing then “Image processing is the analysis and manipulation of a digitized image, especially in order to improve its quality”.

3.4.4 Digital-Image

An image may be defined as a two-dimensional function f(x, y), where x and y are spatial(plane) coordinates, and the amplitude of fat any pair of coordinates (x, y) is called the intensity or grey level of the image at that point.

In another word An image is nothing more than a two-dimensional matrix (3-D in case of coloured images) which is defined by the mathematical function f(x, y) at any point is giving the pixel value at that point of an image, the pixel value describes how bright that pixel is, and what colour it should be. Image processing is basically signal processing in which input is an image and output is image or characteristics according to requirement associated with that image.

Image processing basically includes the following three steps : Importing the image Analysing and manipulating the image Output in which result can be altered image or report that is based on image analysis Applications of Computer Vision:

Here we have listed down some of major domains where Computer Vision is heavily used. Robotics Application

3.4.5 Keras

Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible.

It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), and its primary author and maintainer is François Chollet, a Google engineer. Chollet also is the author of the XCeption deep neural network model.

Features: Keras contains numerous implementations of commonly used neuralnetwork building blocks such as layers, objectives, activation functions, optimizers, anda host of tools to make working with image and text data easier to simplify the 11 coding necessary for writing deep neural network code. The code is hosted on GitHub, and community support forums include the GitHub issues page, and a Slack channel.

Model weights are large file so we have to download and extract the feature from ImageNet database. Some of the popular pre-trained models are listed below,

- ResNet
- VGG 16
- Mobile Net
- InceptionResNetV2
- Inception V3

3.4.6 Numpy

NumPy (pronounced /ˈnʌmpaɪ/ (NUM-py) or sometimes /ˈnʌmpi/ (NUM-pee)) is a library for the Python programming language, adding support for large, multidimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, 12 was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications.

NumPy is opensource software and has many contributors. Features: NumPy targets the CPython reference implementation of Python, which is a non-optimizing bytecode interpreter. Mathematical algorithms written for this version of Python often run much slower than compiled equivalents. NumPy addresses the slowness problem partly by providing multidimensional arrays and functions and operators that operate efficiently on arrays, requiring rewriting some code, mostly inner loops using NumPy.

Using NumPy in Python gives functionality comparable to MATLAB since they are both interpreted,and they both allow the user to write fast programs as long as most operations work on arrays or matrices instead of scalars. In comparison, MATLAB boasts a large number of additional toolboxes, notably Simulink, whereas NumPy is intrinsically integrated with Python, a more modern and complete programming language. Moreover, complementary Python packages are available; SciPy is a library that adds more MATLAB-like functionality and Matplotlib is aplotting package that providesMATLAB-like plotting functionality. Internally, both MATLAB and NumPy rely on BLAS and LAPACK for efficient linear algebra computations. Python bindings of the widely used computer vision library OpenCV utilize NumPy arrays to store and operate on data. Since images with multiple channels are simply represented as three-dimensional arrays, indexing, slicing or masking with other arrays are very efficient ways to access specific pixels of an image

3.4.7 Neural Networks

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. Neural networks can adapt to changing input; so the network generates the best possible resultwithout needing to redesign the output criteria.

The concept of neural networks, which has its roots in artificial intelligence, is swiftly gaining popularity in the development of trading systems.

A neural network works similarly to the human brain’s neural network. A “neuron” in a neural network is a mathematical function that collects and classifies information according to a specific architecture. The network bears a strong resemblance to statistical methods such as curve fitting and regression analysis. A neural network contains layers of interconnected nodes. Each node is a perceptron and is similar to a multiple linear regression. The perceptron feeds the signal produced by a multiple linear regression into an activation function that may be nonlinear.

Neural networks are also ideally suited to help people solve complex problems in real-life situations. They can learn and model the relationships between inputs and outputs that are nonlinear and complex; make generalizations and inferences; reveal hidden relationships, patterns and predictions; and model highly volatile data (such as financial time series data) and variances needed to predict rare events (such as fraud detection)

Convolutional neural networks (CNNs) contain five types of layers: input, convolution, pooling, fully connected and output. Each layer has a specific purpose, like summarizing, connecting or activating. Convolutional neural networks have popularized image classification and object detection. However, CNNs have also been applied to other areas, such as natural language processing and forecasting.

Recurrent neural networks (RNNs) use sequential information such as time-stamped data from a sensor device or a spoken sentence, composed of a sequence of terms. Unlike traditional neural networks, all inputs to a recurrent neural network are not independent of each other, and the output for each element depends on the computations of its preceding elements.

CHAPTER 4

SYSTEM DESIGN

4. 1. SYSTEM ARCHITECTURE

It is the action of extracting an image from a source, typically a hardware-based source, for process of image processing. WebCamera is the hardware-based source

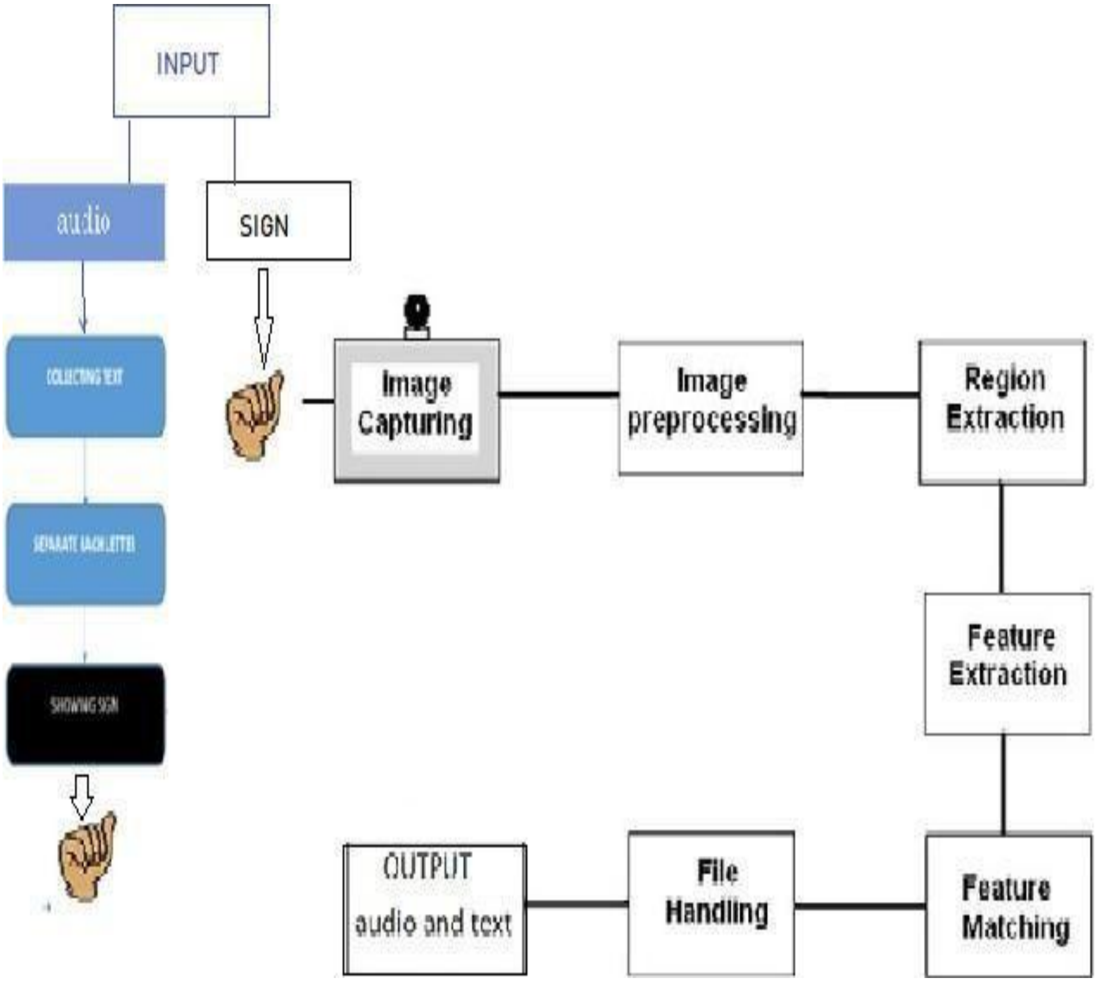


Fig 4.1 Architecture of the proposed system

in our project. It is the first step in the workflow sequence because no processing can be done without an image. The picture that is obtained has not been processed in any way.

The method of separating objects or signs from the context of a captured image is known as segmentation. Proposed Methodology. text subtracting, skin-color detection, and edge detection are all used in the segmentation process. The motion and location of the hand must be detected and segmented in order to recognise gestures.

Predefined features such as form, contour, geometrical feature (position, angle, distance, etc.), colour feature, histogram, and others are extracted from the preprocessed images and used later for sign classification or recognition. Feature extraction is a step in the dimensionality reduction process that divides and organizes a large collection of raw data. reduced to smaller, easier-to-manage classes As a result, processing would be simpler. The fact that these massive data sets have a large number of variables is the most important feature. To process these variables, a large amount of computational power is needed. As a result, function extraction aids in the extraction of the best feature from large data sets by selecting and combining variables into functions. reducing the size of the data These features are simple to use while still accurately and uniquely describing the actual data collection.

Each picture frame is preprocessed to eliminate noise using a variety of filters including erosion, dilation, and Gaussian smoothing, among others. The size of an image is reduced when a color image is transformed to grayscale. A common method for reducing the amount of data to be processed is to convert an image to grey scale.

We’ll use classifiers in this case. Classifiers are the methods or algorithms that are used to interpret the signals. Popular classifiers that identify or understand sign language include the Hidden Markov Model (HMM), KNearest Neighbor classifiers, Support Vector Machine (SVM), Artificial Neural Network (ANN), and Principle Component Analysis (PCA), among others. However, in this project, the classifier will be CNN. Because of its high precision, CNNs are used for image classification and recognition. The CNN uses a hierarchical model that builds a network, similar to a funnel, and then outputs a fully-connected layer in which all neurons are connected to each other and the output is processed.

Understanding human behaviour and identifying various postures and body movements, as well as translating them into text.

A CNN uses a system much like a multilayer perceptron that has been designed for reduced processing requirements. The layers of a CNN consist of an input layer, an output layer and a hidden layer that includes multiple convolutional layers, pooling layers, fully connected layers and normalization layers. The removal of limitations and increase in efficiency for image processing results in a system that is far more effective, simpler to trains limited for image processing and natural language processing.

CHAPTER 5

MODULE DESCRIPTION

5.1 SIGN TO AUDIO

5.1.1 Image Acquisition

Image Acquisition is the first step in any image processing system. The general aim of any image acquisition is to transform an optical image (real-world data) into an array of numerical data which could be later manipulated on a computer.

The gestures are captured through the web camera. This OpenCV video stream is used to capture the entire signing duration. The frames are extracted from the stream and are processed as grayscale images with the dimension of 50*50. This dimension is consistent throughout the project as the entire dataset is sized exactly the same.

5.1.2 Hand Region Segmentation and Hand Region Tracking

The captured images are scanned for hand gestures. This is a part of preprocessing before the image is fed to the model to obtain the prediction. The segments containing gestures are made more pronounced. This increases the chances of prediction by many folds.

5.1.3 Hand Posture Recognition

The preprocessed images are fed to the keras CNN model. The model that has already been trained generates the predicted label. All the gesture labels are assigned with a probability. The label with the highest probability is treated to be the predicted label.

5.1.4 Display as text & speech

The model accumulates the recognized gesture to words. The recognized words are converted into the corresponding speech using the pyttsx3 library. The text to speech result is a simple work around but is an invaluable feature as it gives a feel of an actual verbal conversation which is needed for the project.

5.2 AUDIO TO SIGN LANGUAGE

5.2.1 Forms of Input

Our project is intended to get inputs in multiple formats. The inputs can be of forms:

- Live speech input
- Recorded audio file input

5.2.2 Speech Recognition

The live speech is received as input from the microphone of our system. This is done using the Python package PyAudio. PyAudio is a Python package that is used to record audio on a variety of platforms. The audio thus received is converted into text using Google Speech Recognizer API. It is an API that helps to convert audio to text by incorporating neural network models. In the input format of giving the audio file, the received audio is translated into text by using this Google Speech Recognizer. For lengthier audio files, the audio is divided into smaller chunks on the basis of the occurrence of silence. The chunks are then passed into the Google Speech Recognizer to efficiently convert into text.

5.2.3 Pre-processing of Text

Image preprocessing are the steps taken to format images before they are used by model training and inference. This includes, but is not limited to, resizing, orienting, and color corrections.

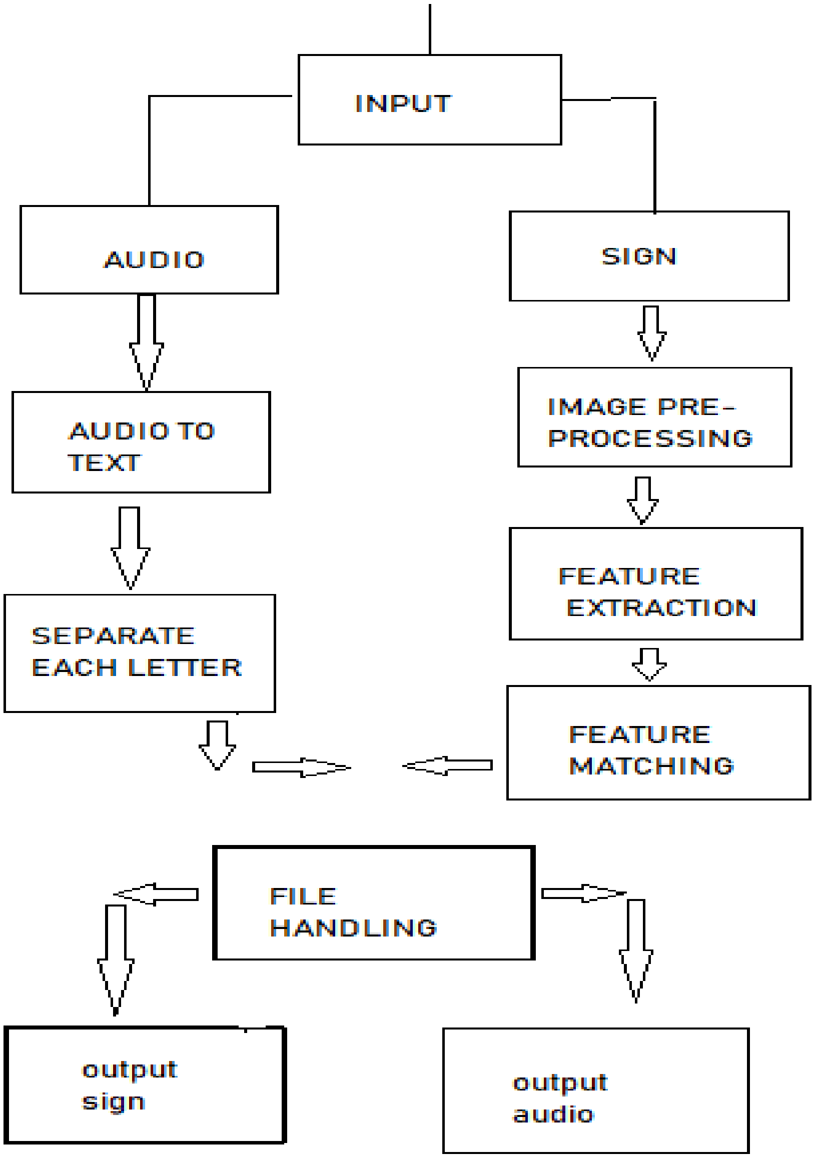


Fig 4.2 Data Flow Diagram

5.2.4 Text to Sign language:

The system iterates through every word in the processed text sentence which is received from the previous step and searches the corresponding sign language video sequences in the local system. If the word is found, the system shows the output as a video sequence using the OpenCV module in Python. OpenCV is an open-source library mainly used for image processing, video analysis, and many more functionalities related to computer vision. The System passes the path of the video sequence to the OpenCV module to play the video. It shows the video sequence frame by frame.

Webscraping is the process of extracting the content from the website. Web scraping is an automatic method to obtain large amounts of data from websites. Most of this data is unstructured data in an HTML format which is then converted into structured data in a spreadsheet or a database so that it can be used in various applications. There are many different ways to perform web scraping to obtain data from websites. These include using online services, particular API's or even creating your code for web scraping from scratch. Many large websites, like Google, Twitter, Facebook, StackOverflow, etc. have API's that allow you to access their data in a structured format. This is the best option, but there are other sites that don't allow users to access large amounts of data in a structured form or they are simply not that technologically advanced. In that situation, it's best to use Web Scraping to scrape the website for data. WebScraping is achieved using the BeautifulSoup module. BeautifulSoup module in Python helps to get or search or navigate or modify the data from the Html files by using parsers.

APPENDIX 1

SAMPLE CODING

```
import speech_recognition as sr
import numpy as np
import matplotlib.pyplot as plt
import cv2
from easygui import *
import os
from PIL import Image, ImageTk
from itertools import count
import tkinter as tk
import string
#import selecting
# obtain audio from the microphone
def func():
        r = sr.Recognizer()
        isl_gif=['any questions', 'are you angry', 'are you busy', 'are you hungry',
'are you sick', 'be careful',
                'can we meet tomorrow', 'did you book tickets', 'did you finish
homework', 'do you go to office', 'do you have money',
                'do you want something to drink', 'do you want tea or coffee', 'do
you watch TV', 'dont worry', 'flower is beautiful',
                'good afternoon', 'good evening', 'good morning', 'good night',
'good question', 'had your lunch', 'happy journey',
```

'hello what is your name', 'how many people are there in your family', 'i am a clerk', 'i am bore doing nothing',

'i am fine', 'i am sorry', 'i am thinking', 'i am tired', 'i dont understand anything', 'i go to a theatre', 'i love to shop',

'i had to say something but i forgot', 'i have headache', 'i like pink colour', 'i live in nagpur', 'lets go for lunch', 'my mother is a homemaker',

'my name is john', 'nice to meet you', 'no smoking please', 'open the door', 'please call me later',

'please clean the room', 'please give me your pen', 'please use dustbin dont throw garbage', 'please wait for sometime', 'shall I help you',

'shall we go together tommorow', 'sign language interpreter', 'sit down', 'stand up', 'take care', 'there was traffic jam', 'wait I am thinking',

'what are you doing', 'what is the problem', 'what is todays date', 'what is your father do', 'what is your job',

'what is your mobile number', 'what is your name', 'whats up', 'when is your interview', 'when we will go', 'where do you stay',

'where is the bathroom', 'where is the police station', 'you are wrong','address','agra','ahemdabad', 'all', 'april', 'assam', 'august', 'australia', 'badoda', 'banana', 'banaras', 'banglore',

'bihar','bihar','bridge','cat', 'chandigarh', 'chennai', 'christmas', 'church', 'clinic', 'coconut', 'crocodile','dasara',

'deaf', 'december', 'deer', 'delhi', 'dollar', 'duck', 'febuary', 'friday', 'fruits', 'glass', 'grapes', 'gujrat', 'hello',

'hindu', 'hyderabad', 'india', 'january', 'jesus', 'job', 'july', 'july', 'karnataka', 'kerala', 'krishna', 'litre', 'mango',

'may', 'mile', 'monday', 'mumbai', 'museum', 'muslim', 'nagpur', 'october', 'orange', 'pakistan', 'pass', 'police station',

```
        'post office', 'pune', 'punjab', 'rajasthan', 'ram', 'restaurant', 'saturday',
        'september', 'shop', 'sleep', 'southafrica',
        'story', 'sunday', 'tamil nadu', 'temperature', 'temple', 'thursday', 'toilet',
        'tomato', 'town', 'tuesday', 'usa', 'village',
        'voice', 'wednesday', 'weight','please wait for sometime','what is your mobile
        number','what are you doing','are you busy']


            arr=['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r',
        's','t','u','v','w','x','y','z']
            with sr.Microphone() as source:
                r.adjust_for_ambient_noise(source)
                i=0
                while True:
                    print("I am Listening")
                    audio = r.listen(source)

try:
                        a=r.recognize_google(audio)
                        a = a.lower()
                        print('You Said: ' + a.lower())

                        for c in string.punctuation:
                          a= a.replace(c,"")

                        if(a.lower()=='goodbye' or a.lower()=='good bye' or
a.lower()=='bye'):

                            print("oops!Time To say good bye")
                    break

                elif(a.lower() in isl_gif):

                  class ImageLabel(tk.Label):
                    """a label that displays images, and plays them if
they are gifs"""
                    def load(self, im):
                      if    isinstance(im,    str):
                        im = Image.open(im)
                      self.loc    =    0
                      self.frames = []

                      try:
                        for i in count(1):

self.frames.append(ImageTk.PhotoImage(im.copy()))
                          im.seek(i)
                      except EOFError:
                        pass


                      try:
                        self.delay = im.info['duration']
                      except:
                        self.delay = 100


                      if len(self.frames) == 1:
                            self.config(image=self.frames[0])
                        else:
                            self.next_frame()

                    def unload(self):
                        self.config(image=None)
                        self.frames = None

                    def next_frame(self):
                        if self.frames:
                            self.loc += 1
                            self.loc %= len(self.frames)
                            self.config(image=self.frames[self.loc])
                            self.after(self.delay, self.next_frame)
            root = tk.Tk()
            lbl = ImageLabel(root)
            lbl.pack()
            lbl.load(r'ISL_Gifs/{0}.gif'.format(a.lower()))
            root.mainloop()
        else:
            for i in range(len(a)):
                            if(a[i] in arr):


                                    ImageAddress = 'letters/'+a[i]+'.jpg'
                                    ImageItself =
Image.open(ImageAddress)
```

```
                             ImageNumpyFormat =
np.asarray(ImageItself)
                             plt.imshow(ImageNumpyFormat)
                             plt.draw()
                             plt.pause(0.8)
                         else:
                             continue

            except:
                print(" ")
            plt.close()
while 1:
 image   = "hand-peace.png"
 msg="HEARING IMPAIRMENT ASSISTANT"
 choices = ["Live Voice","All Done!"]
 reply   = buttonbox(msg,image=image,choices=choices)
 if reply ==choices[0]:
    func()
 if reply == choices[1]:
    quit()\
```

LIVE CAM

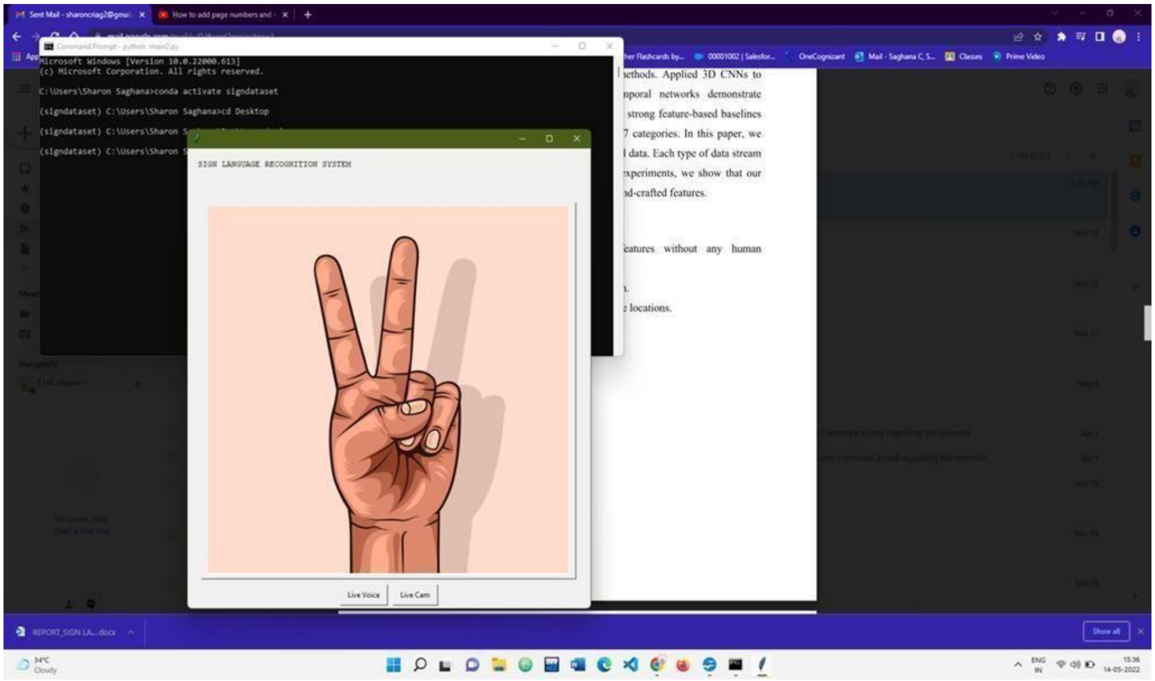
```
import pyttsx3
import numpy as np
import cv2
#import keras
import tensorflow
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
model = tensorflow.keras.models.load_model(r"C:\Users\Sharon
Saghana\Desktop\sign\best_model.h5")
#word_dict =
{6:"call",0:"Doctor",1:"Help",2:"Hot",3:"Lose",4:"Pain",5:"Theif"}
word_dict = {0:'Come',1:'I live in chennai',2:'I am eating',3:'I am working on
it',4:'I am free, this evening',5:'Good afternoon',6:'Good Morning',7:'Lets
begin',8:'Welcome',9:'Okay',10:'Thank you',11:'Done',12:'Sorry',13:'Any
questions?',14:'P',15:'Q',16:'R',17:'U',18:'V',19:'W',20:'Y',21:'Z'}
background = None
accumulated_weight = 0.5
ROI_top = 100         #10
ROI_bottom = 300       #350
ROI_right = 150        #10
ROI_left = 350         #350
def cal_accum_avg(frame, accumulated_weight)
    global background

    if background is None:
        background = frame.copy().astype("float")
        return None
    cv2.accumulateWeighted(frame, background, accumulated_weight)
def segment_hand(frame, threshold=25):
    global background
    diff = cv2.absdiff(background.astype("uint8"), frame)

    _ ,thresholded = cv2.threshold(diff, threshold,
255,cv2.THRESH_BINARY)

     #Fetching contours in the frame (These contours can be of hand
#or any other object in foreground) …
    contours, hierarchy =cv2.findContours(
thresholded.copy(),cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMP
LE)
    # If length of contours list = 0, means we didn't get any
   # contours...
    if len(contours) == 0:
        return None

    else:
        # The largest external contour should be the hand
        hand_segment_max_cont = max(contours, key=cv2.contourArea)

        # Returning the hand segment(max contour) and the
 # thresholded image of hand...
        return (thresholded, hand_segment_max_cont)
def play():
    converter.say(words)
    converter.runAndWait()
cam = cv2.VideoCapture(0)
num_frames =0
converter = pyttsx3.init()
while True:
    ret, frame = cam.read()
    # frame...

    frame = cv2.flip(frame, 1)
    frame_copy = frame.copy()
    # ROI from the frame
    roi = frame[ROI_top:ROI_bottom, ROI_right:ROI_left]
    gray_frame = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)
    gray_frame = cv2.GaussianBlur(gray_frame, (9, 9), 0)
    if num_frames < 70:

        cal_accum_avg(gray_frame, accumulated_weight)

        cv2.putText(frame_copy, "FETCHING BACKGROUND...PLEASE WAIT", (80, 400), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0,0,255), 2)
    else:
        # segmenting the hand region
        hand = segment_hand(gray_frame)

        # Checking if we are able to detect the hand...
        if hand is not None:
            thresholded, hand_segment = hand
            # Drawing contours around hand segment
            cv2.drawContours(frame_copy, [hand_segment + (ROI_right, ROI_top)], -1, (255, 0, 0),1)

            cv2.imshow("Thesholded Hand Image", thresholded)
            thresholded = cv2.resize(thresholded, (64, 64))
            thresholded = cv2.cvtColor(thresholded,cv2.COLOR_GRAY2RGB)
            thresholded =
np.reshape(thresholded,(1,thresholded.shape[0],thresholded.shape[1],3))
            pred = model.predict(thresholded)
            cv2.putText(frame_copy, word_dict[np.argmax(pred)],(170, 45),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,255), 2)
            words = ""
            words = word_dict[np.argmax(pred)]
      print(words)
     #if words:
     #    play()
    # Draw ROI on frame_copy
    cv2.rectangle(frame_copy, (ROI_left, ROI_top), (ROI_right,
    ROI_bottom), (255,128,0), 3)
    # incrementing the number of frames for tracking
    num_frames += 1
    # Display the frame with segmented hand
    cv2.putText(frame_copy, "hand sign recognition",
    (10, 20), cv2.FONT_ITALIC, 0.5, (51,255,51), 1)
    cv2.imshow("Sign Detection", frame_copy)
    print()
    # Close windows with Esc
    k = cv2.waitKey(1) & 0xFF
    if k == 27:
        break
```

```
# Release the camera and destroy all the windows
cam.release()
cv2.destroyAllWindows()
```

APPENDIX 2

SCREENSHOTS



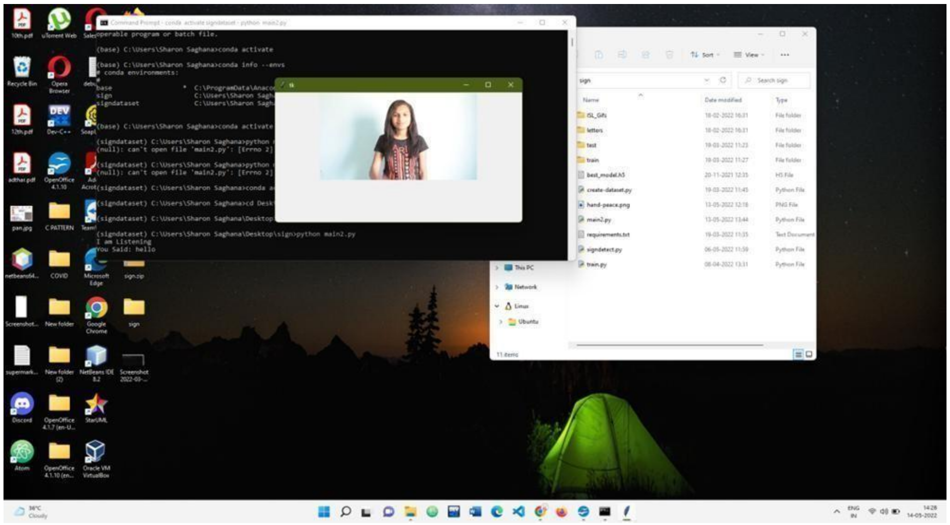
```
Microsoft Windows [Version 10.0.22000.613]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sharon Saghana>conda activate signdataset

(signdataset) C:\Users\Sharon Saghana>cd Desktop

(signdataset) C:\Users\Sharon S

(signdataset) C:\Users\Sharon S
```

SIGN LANGUAGE RECOGNITION SYSTEM

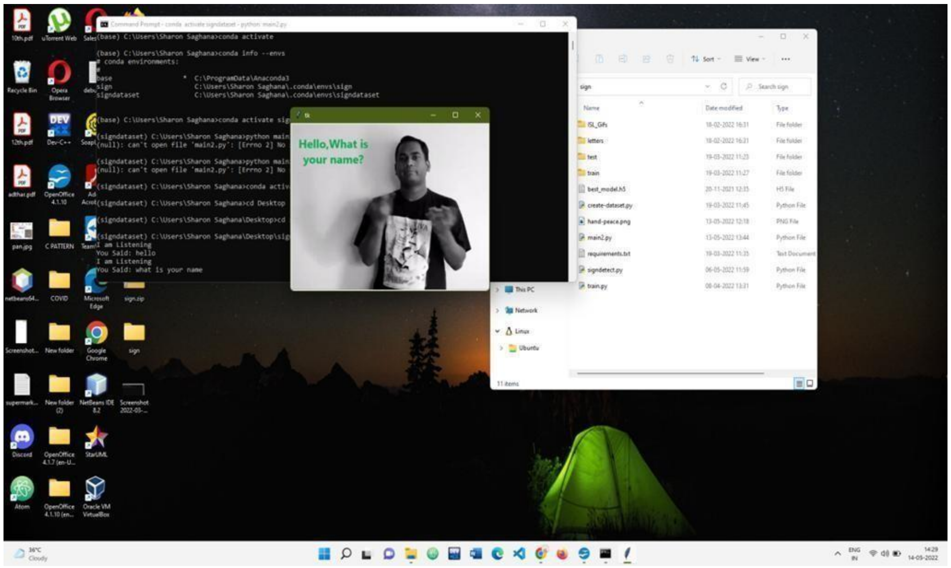
ethods. Applied 3D CNNs to
nporal networks demonstrate
strong feature-based baselines
7 categories. In this paper, we
l data. Each type of data stream
xperiments, we show that our
nd-crafted features.

eatures without any human

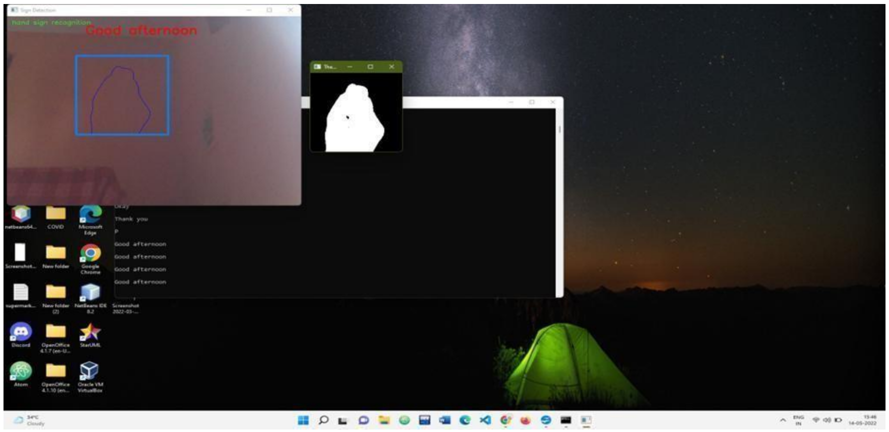
n.
e locations.


Command Prompt - conda activate signdataset - python main2.py
tk

```
operable program or batch file.

(base) C:\Users\Sharon Saghana>conda activate

(base) C:\Users\Sharon Saghana>conda info --envs
# conda environments:
#
base                  *  C:\ProgramData\Anaco
sign                     C:\Users\Sharon Sagh
signdataset              C:\Users\Sharon Sagh

(base) C:\Users\Sharon Saghana>conda activate

(signdataset) C:\Users\Sharon Saghana>python
(null): can't open file 'main2.py': [Errno 2]

(signdataset) C:\Users\Sharon Saghana>python
(null): can't open file 'main2.py': [Errno 2]

(signdataset) C:\Users\Sharon Saghana>conda a

(signdataset) C:\Users\Sharon Saghana>cd Desk

(signdataset) C:\Users\Sharon Saghana\Desktop

(signdataset) C:\Users\Sharon Saghana\Desktop\sign>python main2.py
I am Listening
You Said: hello
```


Command Prompt - conda activate signdataset - python main2.py
tk
Hello,What is your name?

```
(base) C:\Users\Sharon Saghana>conda activate

(base) C:\Users\Sharon Saghana>conda info --envs
# conda environments:
#
base                  *  C:\ProgramData\Anaconda3
sign                     C:\Users\Sharon Saghana\.conda\envs\sign
signdataset              C:\Users\Sharon Saghana\.conda\envs\signdataset

(base) C:\Users\Sharon Saghana>conda activate sig

(signdataset) C:\Users\Sharon Saghana>python main
(null): can't open file 'main2.py': [Errno 2] No

(signdataset) C:\Users\Sharon Saghana>python main
(null): can't open file 'main2.py': [Errno 2] No

(signdataset) C:\Users\Sharon Saghana>conda activ

(signdataset) C:\Users\Sharon Saghana>cd Desktop

(signdataset) C:\Users\Sharon Saghana\Desktop>cd

(signdataset) C:\Users\Sharon Saghana\Desktop\sig
I am Listening
You Said: hello
I am Listening
You Said: what is your name
```



```
Okay
Thank you
P
Good afternoon
Good afternoon
Good afternoon
Good afternoon
```

REFERENCES

[1] Neha Poddar, Shrushti Rao, Shruti Sawant, Vrushali Somavanshi, Prof. Sumita Chandak "Study of Sign Language Translation using Gesture Recognition", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, Issue 2, 2015.

[2] Sawant Pramada, Deshpande Saylee, Nale Pranita, Nerkar Samiksha, Mrs.Archana S, Vaidya "Intelligent Sign Language Recognition Using Image Processing", IOSR Journal of Engineering, Vol. 3, Issue 2, 2013.

[3] Prof. Supriya Pawar, Sainath Muluk, Sourabh Koli "Real Time Sign Language Recognition using Python", International Journal of Innovative Research in Computer and Communication Engineering ,Vol. 6, Issue 3, 2018.

[4] P.V.V.Kishore, "Segment, Track, Extract, Recognize and Convert Sign Language Videos to Voice/Text", International Journal of Advanced Computer Science and Applications, Vol. 3, Issue 6, 2012.

[5] Kamal Preet Kour, Dr. Lini Mathew" Sign Language Recognition Using Image Processing", International Journals of Advanced Research in Computer Science and Software Engineering, Vol. 7, Issue 8, 2017.

[6] Zhi-hua Chen, Jung-Tae Kim, Jianning Liang, Jing Zhang and Yu-Bo Yuan "Real-Time Hand Gesture Recognition Using Finger Segmentation", The Scientific World Journal, Volume 2014, Article ID 267872, 9

[7] Neelam K. Gilorkar, Manisha M. Ingle, “Real Time Detection And Recognition Of Indian And American Sign Language Using Sift”, International Journal of Electronics and Communication Engineering & Technology, Vol. 5, Issue 5, 2014.

[8] T. Ayshee, S. Raka, Q. Hasib, Md. Hossian, R. Rahman, "Fuzzy Rule-Based Hand Gesture Recognition for Bangali Characters", in IEEE International Advanced Computing Conference, 2014.

[9] B. Bauer,H. Hienz "Relevant features for video-based continuous sign language recognition", IEEE International Conference on Automatic Face and Gesture Recognition, 2002.

[10] Patil, Prajakta M., and Y. M. Patil, "Robust Skin Colour Detection and Tracking Algorithm", International Journal of Engineering Research and Technology. Vol.1, Issue8, 2012.