SignoSpeak: Bridging the Gap

Submitted in partial fulfillment of the requirements

of the degree of

Bachelors of Engineering

by

Utsav Kuntalwad (201P049)

Srushti Sawant (201P037)

Mayur Kyatham (201P013)

Prerna Shakwar (201P041)

Guide:

Dr. Varsha Shah

Department of Computer Engineering

Rizvi College of Engineering

University of Mumbai

2023 – 2024

CERTIFICATE

This is to certify that the project entitled **"SignoSpeak: Bridging the Gap"** is a bonafide work of **"Utsav G. Kuntalwad (201P049), Srushti S. Sawant (201P037), Mayur R. Kyatham (201P013), Prerna S. Shakwar (201P041)"** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"Bachelor of Engineering"** in **"Computer Engineering"**.

Dr. Varsha Shah
Internal Guide

(Name and sign)
External Guide / Co-Guide

Shiburaj Pappu
Dean (Academics)

Dr. Anupam Choudhary
H.o.D (Computer)

Dr. Varsha Shah
Principal

Project Report Approval for B.E.

This Project report entitled "**SignoSpeak: Bridging the Gap**" by **Utsav G. Kuntalwad, Srushti S. Sawant, Mayur R. Kyatham and Prerna S. Shakwar** is approved for the degree of Bachelor of Computer Engineering.

Examiners

1. _____

2. _____

Guide

1. _____

2. _____

Date:

Place:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited or from whom proper permission has not been taken when needed.

_____

Utsav Kuntalwad (201P049)

_____

Srushti Sawant (201P037)

_____

Mayur Kyatham (201P013)

_____

Prerna Shakwar (201P041)

Date:

Abstract

SignoSpeak is an innovative application aimed at bridging communication barriers by leveraging advanced technology to detect and interpret fingerspelling based American Sign Language (ASL) gestures. Built on Convolutional Neural Networks (CNNs), SignoSpeak employs cutting-edge computer vision algorithms to analyze live video input from a webcam, accurately recognizing and translating ASL gestures into corresponding text in real-time. This groundbreaking approach enables seamless interaction between individuals proficient in ASL and those unfamiliar with sign language, fostering inclusive communication and breaking down language barriers. The foundation of SignoSpeak lies in its robust CNN architecture, meticulously trained on a diverse dataset of ASL gestures to ensure high accuracy and reliability in gesture recognition. By harnessing the power of deep learning, SignoSpeak is capable of interpreting complex hand movements with precision, offering instant feedback and interpretation on a user-friendly interface. Users can engage in natural conversation using ASL, with SignoSpeak providing real- time translations of detected gestures into easily understandable text. Implemented in Python, SignoSpeak integrates seamlessly with popular libraries such as OpenCV, TensorFlow, and Keras to deliver a responsive and intuitive user experience. Its graphical interface showcases live video feeds alongside recognized ASL symbols and translated text, empowering users to communicate effectively across language barriers. SignoSpeak represents a significant advancement in inclusive technology, catering to individuals with hearing impairments and those seeking to learn sign language. By providing a platform for real-time ASL interpretation, SignoSpeak promotes accessibility and inclusivity, enabling meaningful interactions in diverse social settings. Through continuous refinement and adaptation, SignoSpeak aims to address the evolving needs of its users, paving the way for a more inclusive and accessible society. Our method provides 95.7 % accuracy for the 26 letters of the alphabet.

Keywords: **SignoSpeak, American Sign Language (ASL), Convolutional Neural Networks (CNNs), Computer Vision, Gesture recognition, Real-time translation, Accessibility, Inclusivity, Deep Learning, Communication barriers.**

Index

Figure Index

Chapter-1

Introduction

Breaking barriers in communication, our project, titled 'SignoSpeak: Bridging the Gap,' addresses the fundamental challenge faced by the hearing-impaired community: effective interaction through sign language. In today's increasingly interconnected world, effective communication is essential for fostering understanding and inclusivity among diverse communities. However, for individuals who are deaf or hard of hearing, communication barriers can pose significant challenges, hindering their ability to fully engage in social interactions and access essential services. American Sign Language (ASL) serves as a primary means of communication for millions of individuals worldwide, offering a rich and expressive language through hand gestures, facial expressions, and body movements. Despite its importance, the lack of widespread proficiency in ASL among the general population often leads to misunderstandings, isolation, and limited access to information for deaf individuals.

Addressing these communication barriers requires innovative solutions that leverage technology to facilitate seamless interaction between ASL users and non-signing individuals. This is where SignoSpeak emerges as a pioneering application, designed to bridge the gap between sign language and spoken language through advanced computer vision and deep learning techniques. SignoSpeak aims to empower both ASL users and non-signing individuals by providing real-time interpretation of ASL gestures into easily understandable text, enabling effective communication in diverse social settings.

Our approach transcends mere interpretation; it encapsulates inclusivity and accessibility [2]. Through an intuitive user interface, our system facilitates bidirectional communication, allowing both hearing-impaired individuals and non-signers to engage effortlessly.

Sign language is a visual language and consists of various components. In this project we have targeted recognizing the Hand shape / fingerspelling [4].

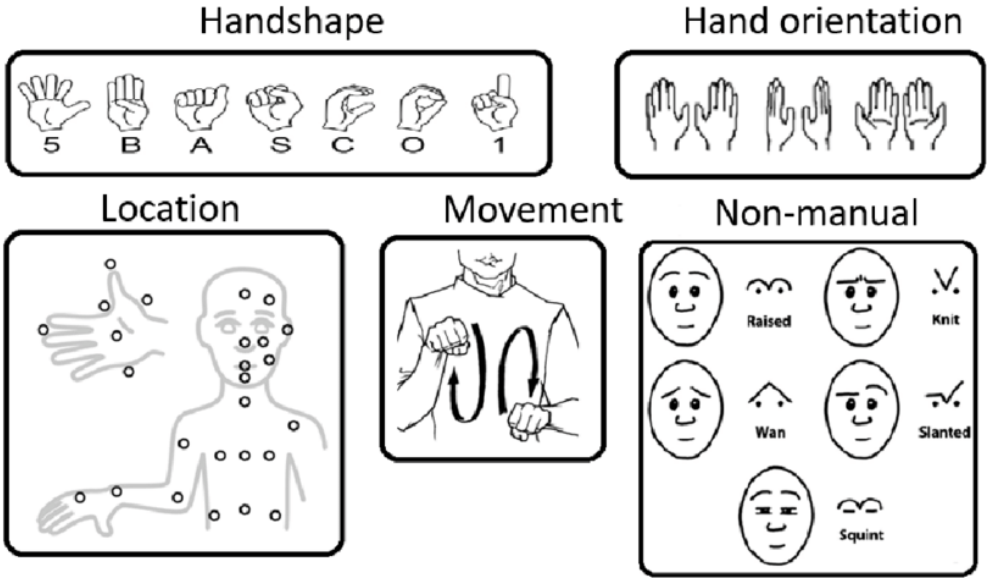


Fig. 01. Components of signs in Sign Language

The primary aim of SignoSpeak is to democratize communication by making ASL more accessible and understandable to a broader audience. By harnessing the capabilities of Convolutional Neural Networks (CNNs), SignoSpeak can accurately recognize and interpret ASL gestures from live video input, offering instant translations into text format. This real-time translation capability facilitates natural and inclusive conversations between ASL users and non-signing individuals, eliminating communication barriers and fostering meaningful interactions [1].

The scope of SignoSpeak extends beyond mere gesture recognition; it encompasses a comprehensive solution for ASL interpretation, incorporating user-friendly interfaces, intuitive controls, and responsive feedback mechanisms[6]. Through continuous refinement and adaptation, SignoSpeak aims to cater to the evolving needs of its users, providing a seamless communication experience that promotes inclusivity and accessibility in various social and professional contexts.

The significant contributions of SignoSpeak lie in its potential to transform the way we perceive and engage with sign language, offering a pathway to greater understanding, empathy, and inclusivity. By facilitating effective communication between ASL users and non-signing individuals, SignoSpeak opens doors to new opportunities for collaboration, education, and social integration [3]. As an innovative application at the intersection of technology and accessibility, SignoSpeak represents a step forward in creating a more inclusive society where communication barriers are minimized, and everyone has the opportunity to be heard and understood.

The gestures we aim to train are as given in the image below.

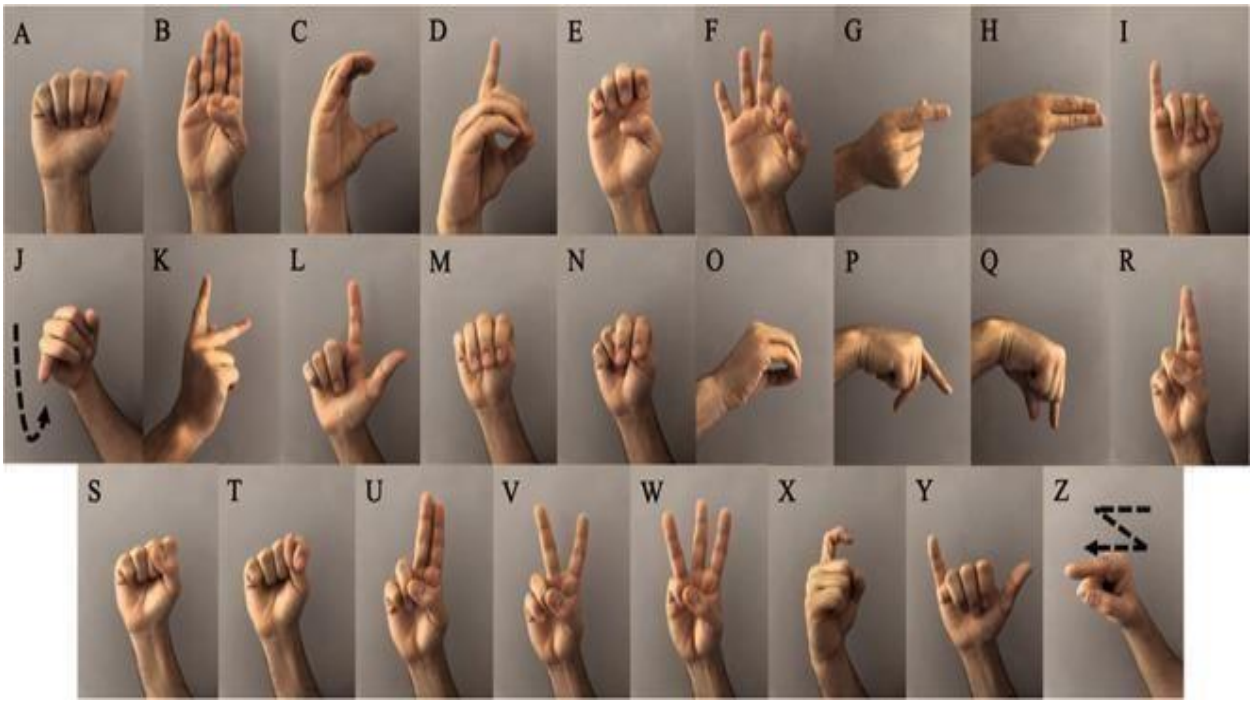


Fig. 02. ASL Hand Gestures for Alphabet

Chapter-2

Literature Survey

Research Paper-01:

Conversion of Sign Language into Text Using Machine Learning Technique [2]

Abstract:
Communication is giving, receiving or exchanging ideas, information, signals or messages through appropriate media, to give information or to express emotions. It is very important and basic need for human beings. But there are some people in our society who born dumb and deaf or deaf due to some medical issues. These people faced many challenges while communicating with each other and also to the normal people. Sign language is one of the commonly used methods by these people for communication. For this translator is required for communication between the person who knows the sign language and to whom they want convey their message. But for many instances' translator is not available which creates a communication gap. This can be overcome with the help of using Machine learning algorithm. The main aim of this work is to provide such system which will convert the hand gestures into text using Convolutional Neural Network (CNN) algorithm.

Methodology:
The proposed recognition technique relies on a convolutional neural network model (CNN) with a feature mapped output layer. The input is captured through the camera using OpenCV and passed through the CNN classifier. In the classifier, the captured input is initially taken and passed through the convolution layer. The number of convolution layer and pooling layer can be increased for more accuracy. After passing through several convolution and pooling layers the output is flattened into a vector and sent into a fully connected layer. The last layer is the SoftMax layer. This layer is used to show the output.

CNN Model: CNN model takes image as an input, process it and classify it under certain categories. In CNN the input images are passed through several layers like the convolution layer, pooling layer, fully connected layers which consists of activation functions.

Python Libraries
1. OpenCV: OpenCV stands for Open Computer Vision which is a library in Python used for Image Processing. This will be required to perform various actions and processing of the captured images.

2. Matplotlib library: Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.

3. NumPy: NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Limitations and Scope:
Sign language is the basic communication method used by hearing impaired people. Hand gestures are used for communication purpose and these people face problems in communicating with other people without a translator. The proposed system aims to fill the communication gap using machine learning by creating a system in which hand gesture can be converted into text using CNN algorithm. Proposed a conversion of sign language into text using CNN. This algorithm is particularly used to improve the recognition accuracy under challenging conditions such as a change in scale, rotation and translation. Using large dataset increases the accuracy of the result.

Research Paper-02:

Hand Gesture Recognition for Emoji Prediction [4]

Abstract:
Emojis are ideograms and smileys visual symbols that are used widely in wireless communication. They present rich novel possibilities of representation and interaction as a new modality. They exist in various genres, including hand gestures, human faces, figures, and signs. Hand gestures, which are the most common and intuitive non-verbal means of communication when we are using a computer, and related work, has recently sparked an interest. Hands often appear in images, videos, and their appearances and pose can give important clues about what people are doing. A combination of hand gestures and emojis can communicate and express the message very conveniently. Considering the positive outcomes of image recognition from specific deep learning methods, we suggest an emoji predictor in real-time. This project consists of a hand gesture recognition method and emoji generator using filters to detect hands and Convolutional Neural Network (CNN) for training the model. Here, a database is being created of hand gestures to train the system. The prediction will be focused on the hand movements and capture the changes by preparing for different hand movement positions to the highest degree of precision.

Methodology:
This project focuses on creating a system for real-time hand gesture recognition and emoji prediction using advanced techniques for image processing and deep learning. Here is a summary of the key components and methods involved:

A) Dataset: The project employs a dataset containing eleven emojis, with 1,200 training images for each emoji to ensure robust model training.

B) Overview: Hand detection is achieved by background subtraction, and the resulting hand contours are used to facilitate emoji prediction.

C) Hand Detection: A continuous camera feed monitors real-time hand gestures, ensuring a consistent environment and resizing images to 350x350 for analysis.

D) Image Processing: Data augmentation techniques, such as horizontal flips, are applied to the dataset. Input images from the webcam are resized to 50x50 for compatibility and processed by converting to HSV. Morphological operations like Gaussian Blur, dilation, and closing enhance precision and remove background noise.

E) Gesture Detection: The model employs a Convolutional Neural Network (CNN) architecture for effective feature extraction and image classification. Preprocessing steps include resizing, thresholding, and centering hand images for scale and translation invariance, enabling real-time gesture recognition.

F) Emoji Prediction: A value is assigned to each gesture for image blending to predict the corresponding emoji. Different weights are applied to the images for transparency and blending.

Limitations and Scope:
Real-time gesture recognition has a wide range of applications, including medical use for individuals with physical disabilities and commercial applications in consumer shops and homes. It enables interactions through hand gestures and emoji prediction for tasks like robot control and office or household applications. However, there are several potential issues to address. These include the need for robust classifiers to handle dynamic gestures and the development of complex motion recognition for advanced human-computer interaction systems. Additionally, ensuring privacy and data security, addressing real-time processing demands, and improving accuracy in recognizing intricate gestures are important considerations.

Chapter-3

Primary descriptors and Descriptions

1. Data acquisition

Data acquisition refers to the process of collecting images or videos of sign language gestures. For SignoSpeak, this involves capturing video streams of individuals performing various American Sign Language (ASL) gestures using a camera

High-quality and diverse data acquisition ensures the model's robustness and ability to recognize a wide range of gestures accurately

The different approaches to acquire data about the hand gesture can be done in the following ways:

A. Use of sensory devices

Sensory devices, such as electromechanical devices or depth sensors can be used to provide exact hand configuration, and position [3]. Different glove-based approaches can be used to extract information. These devices enable the collection of real time video streams or images of individuals performing sign language gestures, which serve as input data for the recognition system. Despite their effectiveness, sensory devices have limitations. Depth sensors, while providing depth information, may be sensitive to occlusions or may not accurately capture fine grained hand movements [2]. Additionally, these sensors require appropriate calibration and setup to ensure accurate data acquisition, which can be time consuming and may introduce additional complexities to the system. Furthermore, the reliance on sensory devices for data acquisition means that our project may be limited in environments where the devices are not available or practical to use, potentially hindering its accessibility and deployment in certain settings.

B. Vision based approach

In vision-based methods, the computer webcam is the input device for observing the information of hands and/or fingers. The Vision Based methods require only a camera, thus realizing a natural interaction between humans and computers without the use of any extra devices, thereby reducing cost[5]. These systems tend to complement biological vision by describing artificial vision systems that are implemented in software and/or hardware. The main challenge of vision-based hand detection ranges from coping with the large variability of the human hand's appearance due to a huge number of hand movements, to different skin-color possibilities as well as to the variations in viewpoints, scales, and speed of the camera capturing the scene [1].

2. Data pre-processing

It is a crucial step in our project, here raw input data, such as images or video frames of sign language gestures, undergo various operations to enhance their quality and suitability for subsequent processing. This stage involves several key processes:

1) Image Enhancement: Raw images or video frames may suffer from noise, variations in lighting or other artifacts. Techniques like noise reduction, contrast enhancement and normalization are applied to improve image quality and ensure consistency across different samples
2) Image Segmentation: Sign language gestures often involves hand movements against complex backgrounds. Image segmentation techniques are used to isolate the hand region from the background, facilitating more accurate gesture recognition [4].

3) Normalization: To improve the robustness of the recognition system, input data are often normalized to a standardized format. This may involve resizing images to a consistent resolution, normalizing color channels, or transforming data to a common coordinate system.

3. Feature extraction for vision-based approach

Extracting meaningful features from the preprocessed data is essential for capturing relevant information about sign language gestures. Features may include hand shape, movement trajectories, or spatial-temporal characteristics, which are critical for accurate gesture classification Firstly, we have used AdaBoost face detector to differentiate between faces and hands as they both involve similar skin color. AdaBoost Face Detector is a Machine Learning algorithm used for classification tasks. It is used to recognize human faces. Using an AdaBoost face detector in conjunction with color-based methods enhances the ability to accurately identify and differentiate between faces and hands, particularly in scenarios where they coexist and have similar skin colors. Then, we also extracted necessary image which is to be trained by applying a filter called Gaussian Blur (Gaussian smoothing). The filter can be easily applied using open computer vision (OpenCV) Applying a Gaussian filter to the images which helps in reducing noise and detail while preserving important image features. It is a filtering operation commonly used in image processing to blur or smooth images. It works by convolving the image with a Gaussian kernel which is a 2D matrix of Gaussian values. The size of the kernel and the standard deviation of the Gaussian values[1]. The size of the kernel and the standard deviation of the Gaussian distribution determines the amount of theblur applied to the image.

We tried doing the hand segmentation of an image using color segmentation techniques but skin color and tone is highly dependent on the lighting conditions due to which output, we got for the segmentation we tried to do were no so great. Moreover, we have a huge number of symbols to be trained for our project many of which look similar to each other like the gesture for symbol 'M' and 'N', hence we decided that in order to produce better accuracies for our large number of symbols, rather than segmenting the hand out of a random background we keep background of hand a stable single color so that we don't need to segment it on the basis of skin color. This would help us to get better results.

4. Representation

The representation of an image as a 3D matrix having dimension as of height and width of the image and the value of each pixel as depth (1 in case of Grayscale and 3 in case of RGB). Further, these pixel values are used for extracting useful features using CNN.

5. Gesture classification

Gesture classification is a fundamental component of the SignoSpeak system, responsible for interpreting sign language gestures captured through sensory devices and converting them into corresponding textual representations. This process involves:

1) Model Training: Supervised learning algorithm, such as Convolution Neural Networks (CNNs) are commonly used for gesture classification. These models learn to map extracted features to predefined classes of sign language gestures through the presentation of labeled training data. During training, the model adjusts its internal parameters to minimize the discrepancy between predicted and ground truth labels, effectively learning discriminative patterns for gesture recognition.

2) Classification Once trained, the gesture classification model can predict the class labels of unseen sign language gestures. Give a new input gesture, the model computes a probability distribution over the set of possible classes and selects the most probable class as the predicted gesture. Classification decisions are based on the learned relationships between input features and gesture categories, enabling accurate recognition of a wide range of sign language expressions.

6. Artificial Neural Network (ANN)

Artificial Neural Network is a connection of neurons, replicating the structure of human brain. Each connection of neuron transfers information to another neuron[5]. Inputs are fed into layer of neurons which processes it and transfers to another layer of neurons called as hidden layers. After processing of information through multiple layers of hidden layers, information is passed to finaloutput layer.

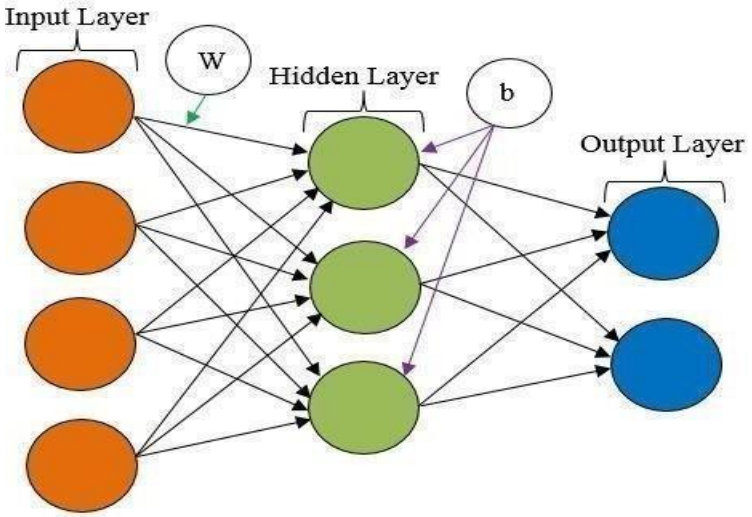


Fig. 03. ANN Architecture

Information flows through the network from the input layer, where data is fed into the network, through the hidden layers, where the processing occurs, to the output layer, where the final result is obtained. Each neuron receives input signals, processes them using a set of weights and biases, and produces an output signal that is transmitted to the neurons in the next layer.

The key concept in ANNs is learning from data[1]. During the training process, the network adjusts its weights and biases based on the difference between the predicted output and the actual output. This is typically done using an optimization algorithm like gradient descent, which minimizes a loss function representing the error between predicted and actual outputs.

Despite their power, ANNs have challenges such as the need for large amounts of labelled data, potential for overfitting, and lack of interpretability. However, they remain a fundamental tool in machine learning and artificial intelligence research and applications.

Various types of ANNs:

1) Feedforward Neural Networks (FNNs)
2) Recurrent Neural Networks (RNNs)
3) Convolutional Neural Networks (CNNs)
4) Generative Adversarial Networks (GANs)
5) Long Short-Term Memory Networks (LSTMs), and so on.

7. Convolutional Neural Network (CNN)

Convolutional Neural Network (**CNN** or **ConvNet**) is a class of **deep neural networks** which is mostly used to do image recognition, image classification, object detection, etc.

Image classification is the task of taking an input image and outputting a class or a probability of classes that best describes the image. In CNN, we take an image as an input, assign importance to its various aspects/features in the image and be able to differentiate one from another[3]. The pre-processing required in CNN is much lesser as compared to other classification algorithms.

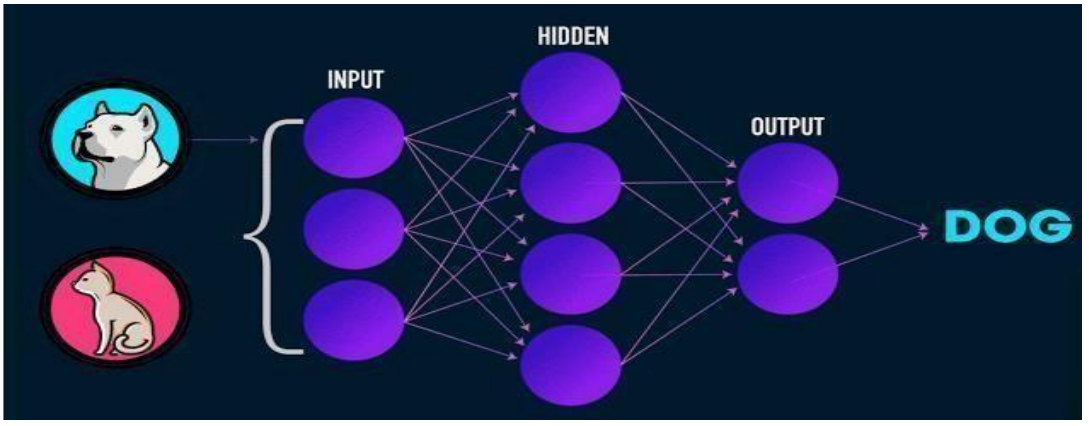


Fig. 04. A classic CNN classifying between a dog and a cat

Architecture:

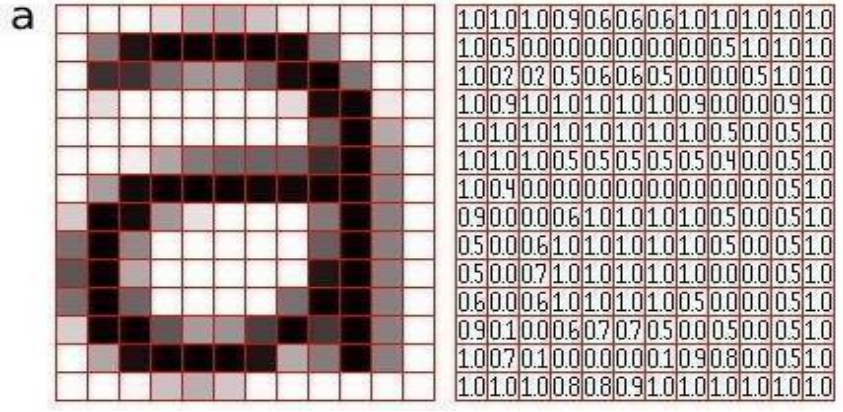


Fig. 05. Matrix Representation of a picture

Computers cannot see things as we do, for computers image is nothing but a matrix.

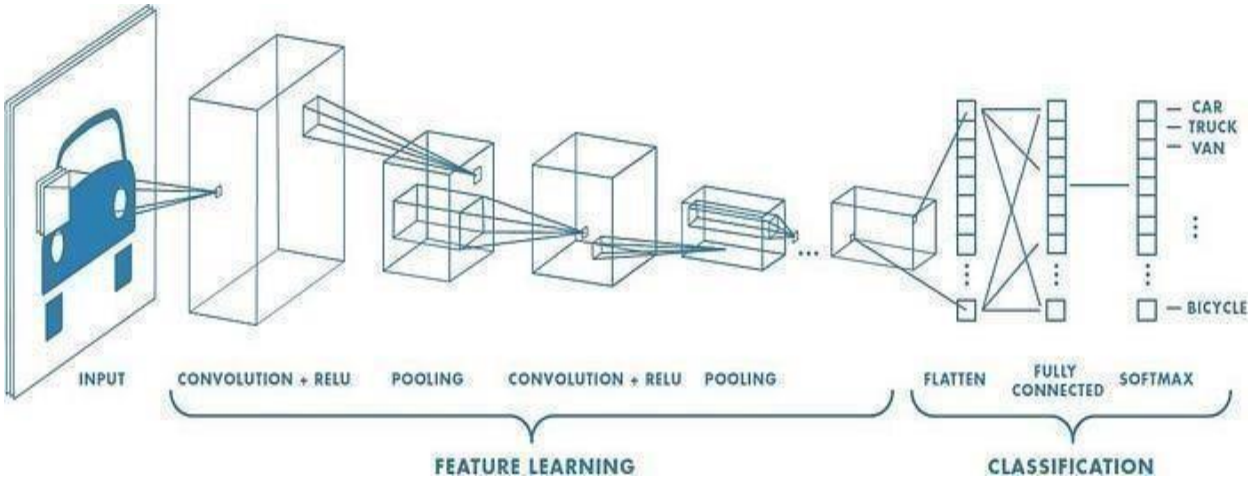


Fig. 06. Different Layers in a CNN

A CNN typically has three layers: a convolutional layer, pooling layer, and fully connected layer.

Convolutional layer:

The main objective of convolution is to extract features such as edges, colors, corners from the input. As we go deeper inside the network, the network starts identifying more complex features such as shapes, digits, face parts as well[4].

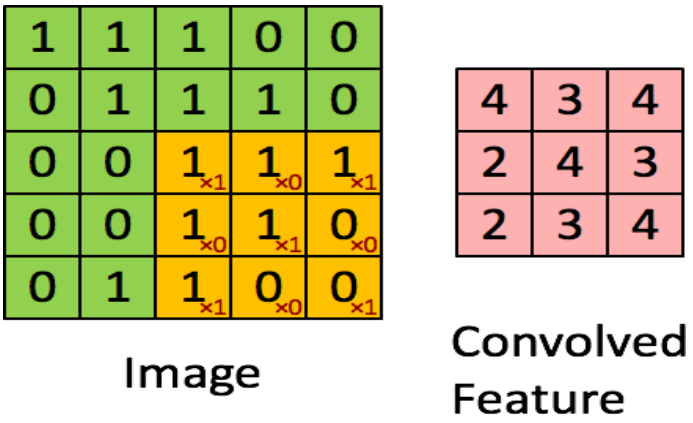


Fig. 07. Convoluting 5x5x1 image with a 3x3x1 kernel to get a 3x3x1 convolved feature

This layer performs a dot product between two matrices, where one matrix (known as filter/kernel) is the set of learnable parameters, and the other matrix is the restricted portion of the image.

If the image is RGB then the filter will have smaller height and width compared to the image but it will have the same depth (height x width x **3**) as of the image.

At the end of the convolution process, we have a **featured matrix** which has lesser parameters(dimensions) than the actual image as well as more clear features than the actual one.

Pooling Layer:

This layer is solely to decrease the computational power required to process the data. It is done by decreasing the dimensions of the featured matrix even more[2]. In this layer, we try to extract the dominant features from a restricted amount of neighborhood.

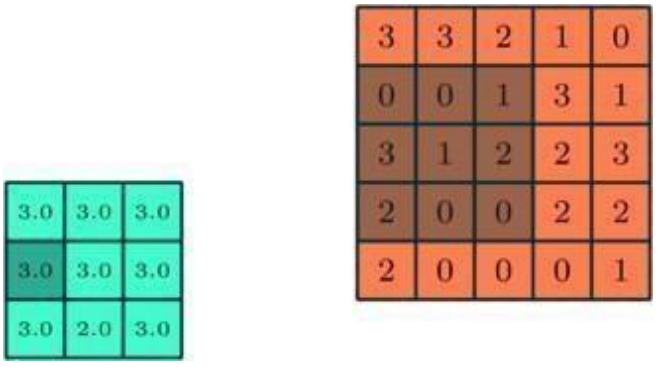


Fig. 08. Pooling layer

The orange matrix is our featured matrix, the brown one is a pooling kernel and we get our blue matrix as output after pooling is done. So, here what we are doing is taking the maximum amongst all the numbers which are in the pooling region and shifting the pooling region each

time to process another neighborhood of the matrix.

There are two types of pooling techniques: **AVERAGE**-*pooling* and **MAX**- *pooling.*

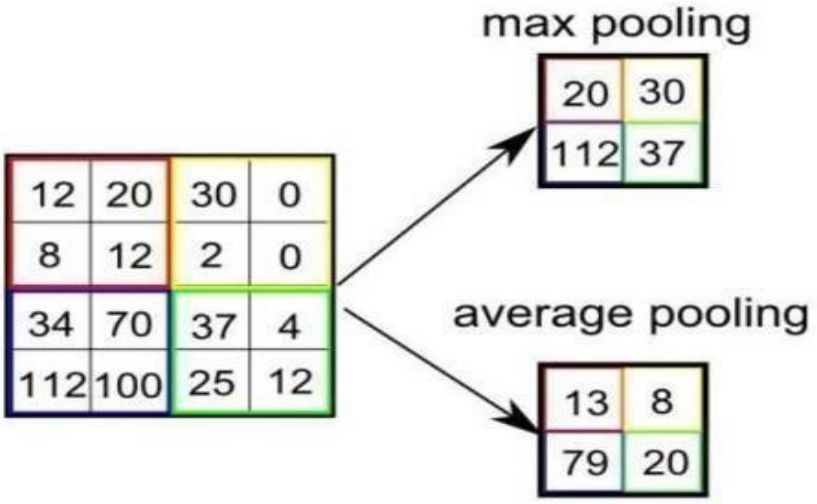


Fig. 09. Max and Average Pooling

The difference between these two is, in **AVERAGE**-*pooling*, we take the average of all the values of pooling region and in **MAX**-*pooling*, we just take the maximum amongst all the values lying inside the pooling region.

So, after pooling layer, we have a matrix containing main features of the image and this matrix has even lesser dimensions, which will help a lot in the next step.

Fully connected layer:

Till now we haven't done anything about classifying different images, what we have done is highlighted some features in an image and reduces the dimensions of the image drastically [2].

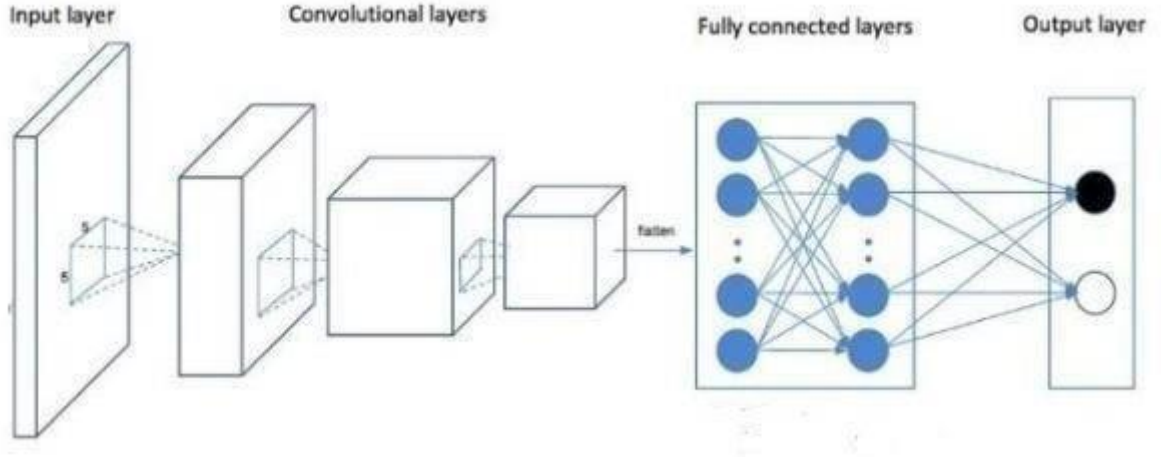


Fig. 10. Fully connected layer inside CNN

From here on, we are actually going to do the classification process.

Now that we have converted our input image into a suitable form for our multi-level fully connected architecture, we shall flatten the image into one column vector. The flattened output is fed to a feed-forward neural network and backpropagation applied to every iteration of training. Over a series of epochs, the model can distinguish between dominating and certain low-level features in images and classify them.

8. TensorFlow

TensorFlow is an open-source machine learning framework developed by Google. It allows developers to build and train various machine learning models efficiently. TensorFlow is renowned for its flexibility, scalability, and extensive library of pre-built modules for tasks such as image and speech recognition, natural language processing, and reinforcement learning. Its core component is the tensor, a multidimensional array, which forms the basis for computations in neural networks. TensorFlow's computational graph abstraction enables seamless distribution of work across multiple CPUs or GPUs, making it ideal for both research and production environments.

9. Keras

Keras is a high-level neural networks library written in python that works as a wrapper to TensorFlow. It is used in cases where we want to quickly build and test the neural network with minimal lines of code. It contains implementations of commonly used neural network elements like layers, objective, activation functions, optimizers, and tools to make working with images and text data easier.

10. OpenCV

OpenCV, or Open-Source Computer Vision Library, is a versatile and widely-used open-source library for computer vision and image processing tasks. It provides a comprehensive suite of functions and algorithms for tasks such as image and video analysis, object detection and tracking, facial recognition, and machine learning integration [5]. Its ease of use, extensive documentation, and cross-platform compatibility make it a popular choice for both academic research and industrial applications in fields ranging from robotics to augmented reality.

It is mainly used for image processing, video capture and analysis for features like face and object recognition.

11. NumPy

NumPy is a powerful Python library for numerical computing that provides support for multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently. It is widely used in scientific computing, data analysis, and machine learning applications due to its speed and versatility. NumPy's core data structure is the `ndarray`, which enables efficient storage and manipulation of large datasets. Additionally, NumPy provides tools for array creation, indexing, slicing, reshaping, and broadcasting, making it essential for numerical operations in Python.

12. Tkinter

Tkinter is a standard GUI (Graphical User Interface) toolkit for Python, providing a platform-independent way to create graphical applications. It is based on the Tk GUI toolkit, which originated as part of the Tcl scripting language[2]. Tkinter allows developers to create windows, dialogs, buttons, menus, and other GUI components to build interactive applications with ease. It provides a simple and intuitive interface for creating user-friendly applications, making it suitable for both beginners and experienced developers. With Tkinter, developers can design attractive and functional GUIs for their Python applications, whether they're desktop utilities, data visualization tools, or interactive software solutions.

Chapter-4

Methodology

The system adopts a vision-based approach, relying solely on bare hands to represent all signs, thereby eliminating the need for artificial devices during interaction.

4.1 Data Set Generation:

Despite efforts to locate pre-existing datasets matching our requirements, only datasets in the form of RGB values were found, prompting the decision to create a bespoke dataset. We utilized the OpenCV library to generate our dataset, capturing approximately 600 - 650 images of each ASL symbol for training and around 200 - 230 images per symbol for testing purposes.

Initially, we captured frames from the webcam, defining a Region of Interest (ROI) marked by a blue bounded square.

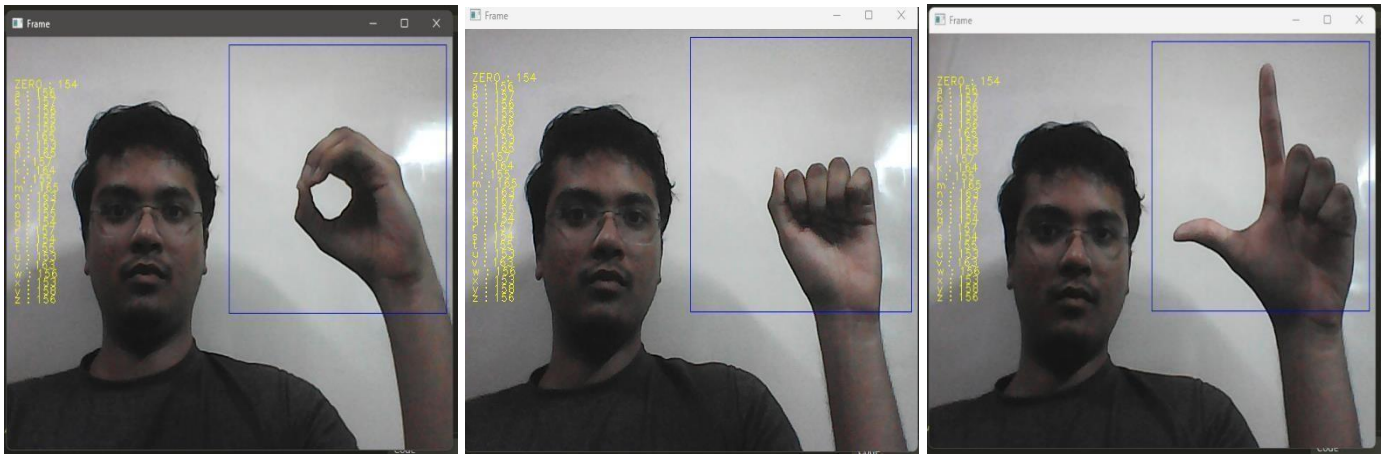


Fig. 11. Creating Dataset for Training and Testing purpose

Following this, a Gaussian Blur Filter was applied to each image to extract various features effectively. The resulting image, post-Gaussian Blur application, facilitated enhanced feature extraction and pre-processing, vital for subsequent implementation steps.

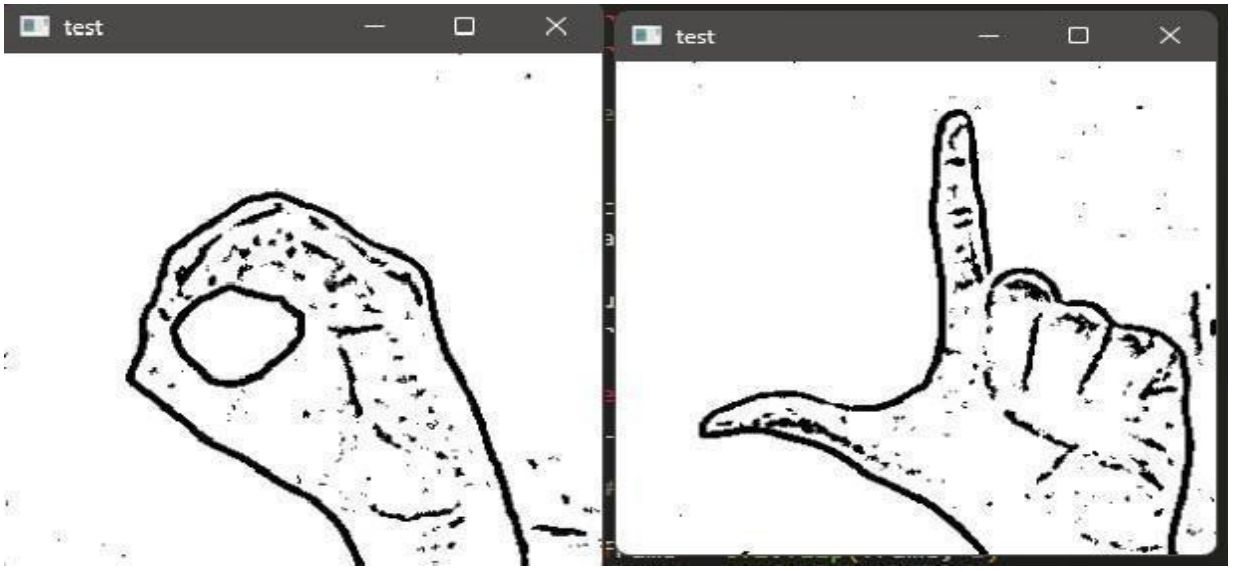


Fig. 12. Gaussian Blur Filter

4.2 Gesture Classification:

Our approach utilizes two layers of algorithms to predict the user's final symbol.

Algorithm Layer 1:

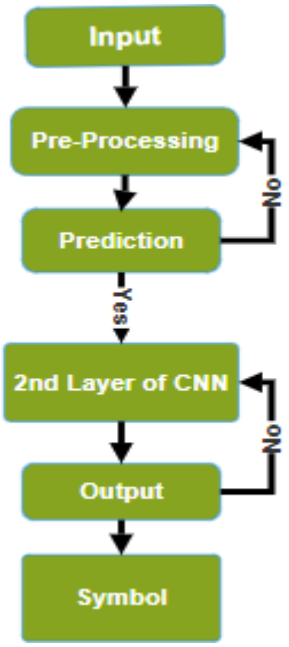


Fig. 13. Flowchart of SignoSpeak

1. The frame captured with OpenCV undergoes processing via Gaussian Blur filter and thresholding to extract features, resulting in a processed image.
2. This processed image is then fed into the CNN model for prediction. If a letter is detected consistently for more than 50 frames, it is printed and considered for word formation[2].
3. The algorithm also accounts for spaces between words, utilizing the blank symbol accordingly.

Algorithm Layer 2:

1. Various sets of symbols exhibiting similar detection results are identified.
2. Classifiers specific to these sets are employed to distinguish between them effectively.

Layer 1:

1. 1st Convolutional Layer: Initially, the input image, with a resolution of 128x128 pixels, undergoes processing through the first convolutional layer employing 32 filter weights (3x3 pixels each). This operation results in a 126x126 pixel image for each filter weight.
2. 1st Pooling Layer: Subsequently, the images are down sampled through max pooling of 2x2, retaining the highest value within each 2x2 array. As a consequence, the image is reduced to a resolution of 64x64 pixels.
3. 2nd Convolutional Layer: The output from the first pooling layer, sized 64x64 pixels, serves as input for the second convolutional layer, processed with 32 filter weights (3x3 pixels each). This yields a 60x60 pixel image.
4. 2nd Pooling Layer: Further down sampling occurs through max pooling of 2x2, reducing the image resolution to 32x32 pixels.
5. 1st Densely Connected Layer: The resulting images are fed into a fully connected layer comprising 128 neurons. The output from the second convolutional layer is reshaped into an array of 32x32x32 = 32768 values. A dropout layer with a dropout rate of 0.5 is incorporated to prevent overfitting.

6. 2nd Densely Connected Layer: The output from the 1st Densely Connected Layer is directed to a fully connected layer with 96 neurons.
7. Final Layer: Finally, the output from the 2nd Densely Connected Layer serves as input for the final layer, which contains neurons equal to the number of classes being classified (alphabets + blank symbol = 27).

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 128, 128, 32)	320
max_pooling2d (MaxPooling2D)	(None, 64, 64, 32)	0
conv2d_1 (Conv2D)	(None, 64, 64, 32)	9,248
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 32)	0
flatten (Flatten)	(None, 32768)	0
dense (Dense)	(None, 128)	4,194,432
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 96)	12,384
dropout_1 (Dropout)	(None, 96)	0
dense_2 (Dense)	(None, 64)	6,208
dense_3 (Dense)	(None, 27)	1,755

Total params: 4,224,347 (16.11 MB)
Trainable params: 4,224,347 (16.11 MB)
Non-trainable params: 0 (0.00 B)

Fig. 14. Model Summary

Activation Function:

ReLU (Rectified Linear Unit) is employed in all layers (convolutional and fully connected neurons). ReLU introduces nonlinearity, facilitating the learning of complex features[5]. It addresses the vanishing gradient problem and accelerates training by reducing computation time.

Pooling Layer:

Max pooling is applied to the input image with a pool size of (2, 2) and ReLU activation function. This reduces parameters, diminishing computation costs and overfitting.

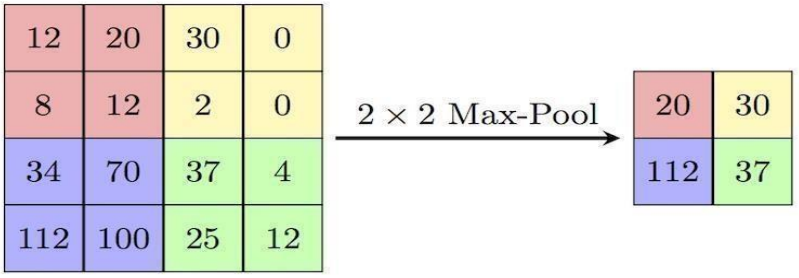


Fig. 15. Max Pooling

Dropout Layers:

To combat overfitting, dropout layers are integrated. These layers randomly deactivate a set of activations, ensuring the network's capability to provide accurate classification despite dropout occurrences.

Optimizer:

Adam optimizer is utilized for updating the model based on the loss function output [3]. Combining thestrengths of ADA GRAD and RMSProp, Adam optimizer optimizes model performance effectively.

Layer 2:

We employ two layers of algorithms to verify and predict symbols that bear close resemblance to each other, thereby enhancing our system's accuracy in symbol detection. During testing, certain symbols exhibited ambiguity and were often misclassified as others:

1. For the symbol 'D': It was frequently mistaken for 'R' and 'U'.
2. For the symbol 'U': It was commonly misinterpreted as 'D' and 'R'.
3. For the symbol 'I': It was often confused with 'T', 'D', 'K', and even itself.
4. For the symbol 'S': It showed similarity with 'M' and 'N'.

To address these challenges, we developed three distinct classifiers tailored to classify these symbol sets effectively:

1. {D, R, U}
2. {T, K, D, I}
3. {S, M, N}

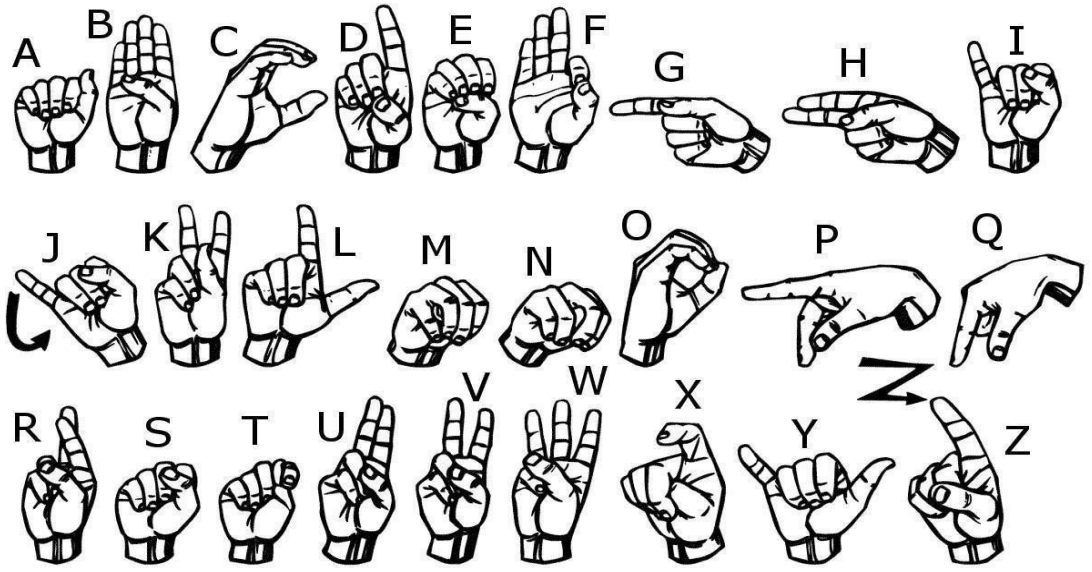


Fig. 16. Similar Sign's in ASL

Training and Testing:

We convert our input images (RGB) into grayscale and apply Gaussian blur to eliminate unnecessary noise. Subsequently, we apply adaptive thresholding to extract our hand from the background and resize our images to 128 x 128 pixels. The pre-processed input images are then fed into our model for both training and testing, following all the aforementioned operations.

The prediction layer estimates the likelihood of the image falling under one of the classes [6]. Consequently, the output is normalized between 0 and 1, ensuring that the sum of each value in each class totals to 1.

Initially, the output of the prediction layer may deviate considerably from the actual value. To improve accuracy, we train the networks using labelled data. Cross-entropy serves as a performance measurement for classification tasks. It is a continuous function that yields positive values when the prediction differs from the labelled value and zeros when they match precisely. Therefore, we optimize the cross-entropy by minimizing it as close to zero as possible. To accomplish this, we adjust the weights of our neural networks in the network layer. TensorFlow provides an inbuilt function to compute the cross-entropy.

Upon identifying the cross-entropy function, we optimize it using Gradient Descent, with the Adam Optimizer proving to be the most effective gradient descent optimizer [1]. This optimization process enhances the model's performance and accuracy in classifying hand gestures effectively.

4.3 Challenges Faced:

SignoSpeak faces several challenges inherent to the domain of sign language recognition and translation. Some of these challenges include:

1. Variability in Signing: Sign language involves a wide range of gestures, handshapes, movements, and facial expressions. The variability in signing styles among different individuals can make it challenging to develop a robust recognition system that works accurately for everyone.

2. Complexity of Gestures: Sign language gestures can be complex, involving intricate hand movements and spatial arrangements. Capturing and interpreting these gestures accurately requires sophisticated algorithms capable of understanding and representing the nuances of sign language.

3. Ambiguity and Homophonous Signs: Some signs in sign language may look similar but have different meanings (homophonous signs). Distinguishing between these signs accurately is crucial for effective communication but can be challenging for a recognition system.

4. Non-Manual Components: Sign language includes not only hand movements but also facial expressions, body posture, and other non-manual components that convey meaning. Incorporating these non-manual components into the recognition system adds complexity and requires advanced computer vision techniques.

5. Real-time Processing: Sign language interpretation often needs to be performed in real-time to facilitate smooth communication between signers and non-signers. Achieving low-latency processing while maintaining high accuracy poses a significant challenge.

6. Data Collection and Annotation: Collecting a large and diverse dataset of sign language samples, along with accurate annotations, is crucial for training robust recognition models. However, data collection can be time-consuming and expensive, especially for sign languages with fewer resources and speakers.

7. User Adaptation: Sign language recognition systems need to be adaptable to different users with varying signing styles, hand shapes, and speeds. Designing systems that can adapt to individual users' preferences and idiosyncrasies is an ongoing challenge.

8. Hardware Limitations: Real-time sign language recognition systems may need to run on resource-constrained devices such as smartphones or wearable devices. Optimizing algorithms for efficiency and minimizing computational requirements without sacrificing accuracy is essential.

Addressing these challenges requires a multidisciplinary approach that combines expertise in computer vision, machine learning, linguistics, human-computer interaction, and sign language studies.

4.4 Relevance to PO & PSO of the department

Relevance to PO (Out of 3)

Engineering knowledge	3
Problem analysis	2
Design/development of solutions	3
Conduct investigations of complex problems	3
Modern tool usage	3
The engineer and society	2
Environment and sustainability	2
Ethics	3
Individual and team work	3
Communication	3
Project management and finance	3
Life-long learning	3

Relevance to PSO (Out of 3)

Open-Source Tools	3
Industry Readiness	3

Chapter-5

Result

In this project, a functional real-time vision-based American Sign Language (ASL) recognition system has been developed for ASL alphabets, catering to the needs of Deaf and Dumb individuals. Through this approach, we are able to detect nearly all the symbols accurately, provided they are displayed properly, without background noise, and with adequate lighting. Initially, we achieved an accuracy of 92% using only layer 1 of our algorithm. However, by integrating both layer 1 and layer 2, we achieved a higher accuracy of 95.7%. This accuracy surpasses that of most current research papers on American Sign Language Recognition.

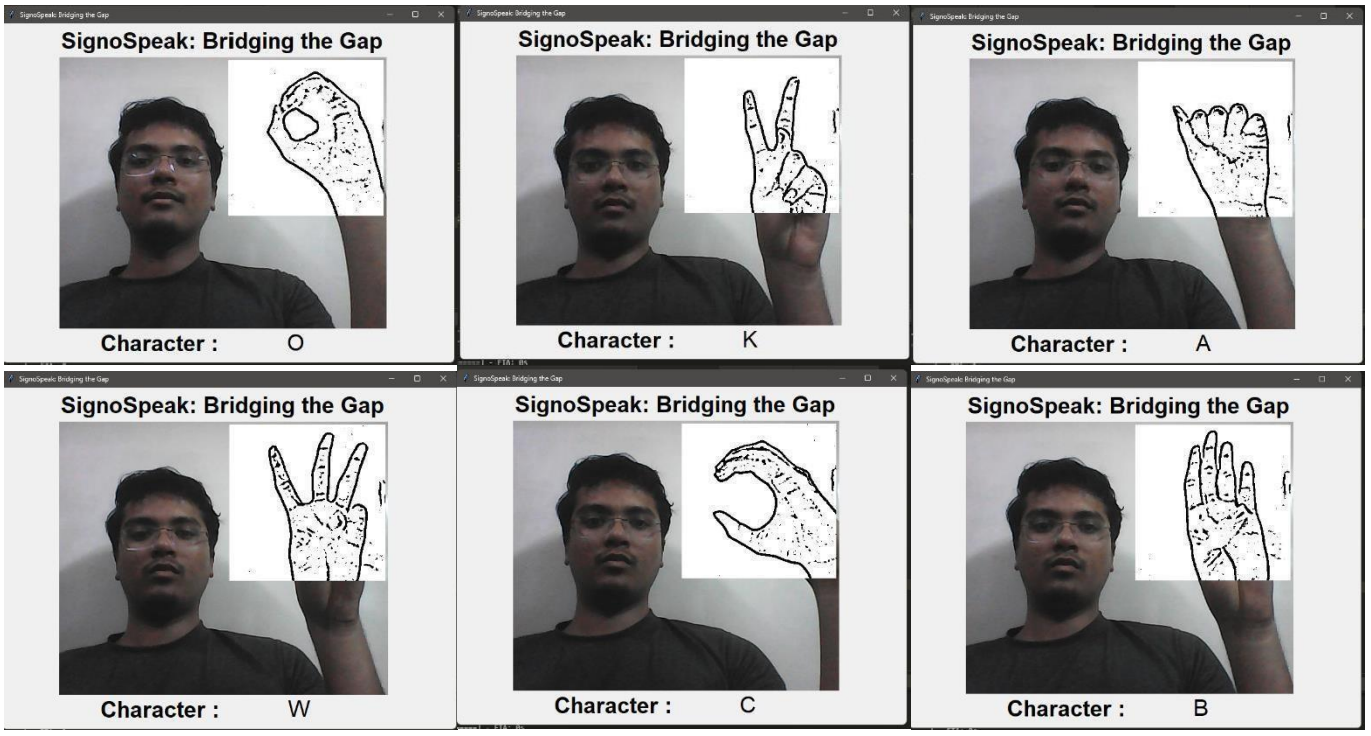


Fig. 17 Sign Language detection

The implementation of SignoSpeak enables real-time sign language recognition, facilitating seamless communication between signers and non-signers. The low-latency processing capabilities ensure swift and efficient translation of sign language gestures into text or speech, enhancing accessibility and inclusivity.

SignoSpeak exhibits adaptability to different signing styles and individual user preferences. The system can generalize well to unseen data and users, demonstrating the effectiveness of the trained CNN models in capturing the underlying patterns and variations in sign language gestures.

While SignoSpeak represents a significant advancement in sign language recognition, there are several avenues for future research and improvement. Areas such as fine-tuning the CNN models for specific sign language dialects, enhancing the system's robustness to environmental factors and occlusions, and integrating natural language processing for more context-aware translations offer promising directions for future work.

Chapter-6

Conclusion

1. Accuracy and Reliability: The SignoSpeak project has demonstrated high accuracy and reliability in recognizing and translating sign language gestures, indicating the effectiveness of the CNN-based approach in capturing and analyzing complex visual patterns.

2. Real-world Viability: Through rigorous testing and validation, it is evident that SignoSpeak is viable for real-world applications, offering practical solutions for bridging the communication gap between signers and non-signers.

3. User Feedback and Satisfaction: User feedback obtained from validation studies reflects a high level of satisfaction with SignoSpeak's performance and usability, underscoring its potential to positively impact the lives of sign language users.

4. Robustness and Generalization: The robustness of SignoSpeak to variations in signing styles and environmental factors signifies its ability to generalize well to diverse scenarios, enhancing its applicability in different contexts.

5. Future Directions: The findings highlight several avenues for further research and improvement, including fine-tuning the CNN models, integrating additional modalities such as facial expressions, and exploring multi-lingual and multi-cultural sign language recognition.

6. Social Impact: SignoSpeak has the potential to contribute significantly to promoting inclusivity and accessibility for the deaf and hard of hearing community, fostering greater understanding and communication between individuals with diverse communication needs.

7. Technological Advancement: The successful implementation of SignoSpeak underscores the role of advanced machine learning techniques, such as CNNs, in addressing complex real-world challenges and advancing assistive technologies.

8. Collaborative Opportunities: Collaboration with stakeholders, including sign language users, educators, and assistive technology experts, can further enhance the development and adoption of SignoSpeak, ensuring its alignment with user needs and preferences.

9. Policy Implications: The adoption of SignoSpeak may necessitate policy changes and advocacy efforts to promote its widespread use and integration into educational, workplace, and public settings, fostering a more inclusive society.

10. Continued Evaluation and Iteration: Continuous evaluation and iteration of SignoSpeak are essential to maintain its effectiveness and relevance, as well as to address emerging challenges and opportunities in the field of sign language recognition and translation.

Chapter-7

References

[1] T. Yang, Y. Xu, and "A., Hidden Markov Model for Gesture Recognition", CMU-RI-TR-94 10, Robotics Institute, Carnegie Mellon Univ., Pittsburgh, PA, May 1994.

[2] Pujan Ziaie, Thomas Muller, Mary Ellen Foster, and Alois Knoll "A Na ıve Bayes Munich, Dept. of Informatics VI, Robotics and Embedded Systems, Boltzmannstr. 3, DE-85748 Garching, Germany.

[3] Mohammed Waleed Kalous, Machine recognition of Auslan signs using PowerGloves: Towards large-lexicon recognition of sign language.

[4] Pigou L., Dieleman S., Kindermans PJ., Schrauwen B. (2015) Sign Language Recognition Using Convolutional Neural Networks. In: Agapito L., Bronstein M., Rother C. (eds) Computer Vision - ECCV 2014 Workshops. ECCV 2014. Lecture Notes in Computer Science, vol 8925. Springer, Cham.

[5] Number System Recognition (https://github.com/chasinginfinity/number-sign-recognition)

[6] Zaki, M.M., Shaheen, S.I.: Sign language recognition using a combination of new vision-based features. Pattern Recognition Letters 32(4), 572–577 (2011).

[7] https://github.com/emnikhil/Sign-Language-To-Text-Conversion

[8] N. Mukai, N. Harada and Y. Chang, "Japanese Fingerspelling Recognition Based on Classification Tree and Machine Learning," 2017 Nicograph International (NicoInt), Kyoto, Japan, 2017, pp. 19-24. doi:10.1109/NICOInt.2017.9

[9] Byeongkeun Kang, Subarna Tripathi, Truong Q. Nguyen" Real-time sign language fingerspelling recognition using convolutional neural networks from depth map" 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)

[10] Mr. G. Sekhar Reddy, A. Sahithi, P. Harsha Vardhan, P. Ushasri, International Journal for Research in Applied Science & Engineering Technology (IJRASET), ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538

[11] Sayali Gore, Namrata Salvi, Swati Singh, "Conversion of Sign Language into Text Using Machine Learning Technique",International Journal of Research in Engineering, Science and Management, Volume 4, Issue 5, May 2021

[12] R Sreemathy, Continuous word level sign language recognition using an expert system based on machine learning, International Journal of Cognitive Computing in Engineering

[13] https://opencv.org/

[14] Journal of Artificial Intelligence Research and Advances ISSN:2395-6720 Vol. 09 No.3 2022

[15] https://en.wikipedia.org/wiki/Convolutional_neural_nework

[16] https://www.ijert.org/a-review-paper-on-sign-language-recognition-for-the-deaf-and-dumb

[17] Mahesh Kumar N B, "Conversion of Sign Language into Text", International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 9 (2018) pp. 7154-7161

[18] Asian Journal of Computer Science and Technology ISSN: 2249-0701 Vol. 11 No.1

[19] https://link.springer.com/chapter/10.1007/978-981-19-9888-1_21

[20]https://ieeexplore.ieee.org/document/9987362

[21] Artificial Intelligence a guide to intelligent systems by Michael Negnevitsk

Acknowledgement

We are profoundly grateful to Dr. Varsha Shah for her expert guidance and continuous encouragement throughout to see that this project rights its target.

We would like to express deepest appreciation towards Prof. Shiburaj Pappu, Dean RCOE, Mumbai and Prof. Anupam Choudhary, HOD of Computer Department whose invaluable guidance supported us in this project.

At last, we would express our sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped us directly or indirectly during this course of work.

Utsav G. Kuntalwad (201P049)
Srushti S. Sawant (201P037)
Mayur R. Kyatham (201P013)
Prerna S. Shakwar (201P041)

iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** IV **Month of publication:** April 2024

DOI: https://doi.org/10.22214/ijraset.2024.59787

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

SignoSpeak: Bridging the Gap

Utsav G. Kuntalwad[1], Srushti S. Sawant[2], Mayur R. Kyatham[3], Prerna S. Shakwar[4], Dr. Varsha Shah[5]
Computer Engineering Department, Mumbai University

Abstract: ***"SignoSpeak: Bridging the Gap" is an innovative software research aimed at transforming communication for the hearing-impaired. It converts sign language to text in real-time, breaking down communication barriers effectively. Besides text conversion, SignoSpeak interprets various gestures, enabling users to express their thoughts vividly. By translating hand and body movements into text and audio, it enriches communication and information conveyance. This device acts as a crucial link between sign language and written language, benefiting those dependent on sign language. Additionally, its ability to recognize gestures fosters inclusive cross-cultural communication. SignoSpeak signifies a significant milestone in promoting inclusivity and meaningful interaction for the hearing-impaired. Its potential lies in revolutionizing global communication norms and understanding the unique needs of individuals with hearing impairments.***

Keywords: CNN (Convolutional Neural Network), Cross-cultural communication, Hand Gestures, Hearing-impaired, Real-time conversion, SignoSpeak

I. INTRODUCTION

American Sign Language (ASL) is essential for Deaf and Dumb (D&M) individuals who cannot use spoken language for communication. ASL employs hand gestures, facial expressions, and body movements to convey meaning effectively. Unlike spoken languages, ASL is not universal and varies by region. The main objective of the research is to translate sign language to text language [3]. Efforts to bridge the communication gap between D&M and non-D&M individuals have become essential for ensuring effective interaction. Sign language is a visual language and consists of three major components:

TABLE I. Major components of visual language

Finger Spelling	World Level Sign Vocabulary	Non-manual features
Used to spell words letter by letter.	Used for the majority of communication.	Facial expressions and tongue, mouth and body position.

Depth sensors enable us to capture additional information to improve accuracy and/or processing time. Also, with recent improvement of GPU, CNNs have been employed to many computer vision problems. Therefore, we take advantage of a depth sensor and convolutional neural networks to achieve a real-time and accurate sign language recognition system [4]. In order to overcome the gap in communication caused by the difference in modes of communication, an interpreter is necessary to reduce the confusion. This research is an attempt to ease the communication between deaf and normal people. Sign language translation, a burgeoning area of research, facilitates natural communication for those with hearing impairments. A hand gesture recognition system provides deaf individuals with the means to communicate with verbal individuals independently, eliminating the need for an interpreter. Our research centres on developing a model that can accurately recognize Fingerspelling-based hand gestures, enabling the formation of complete words through the combination of each gesture. The gestures we aim to train are as given in the image below.

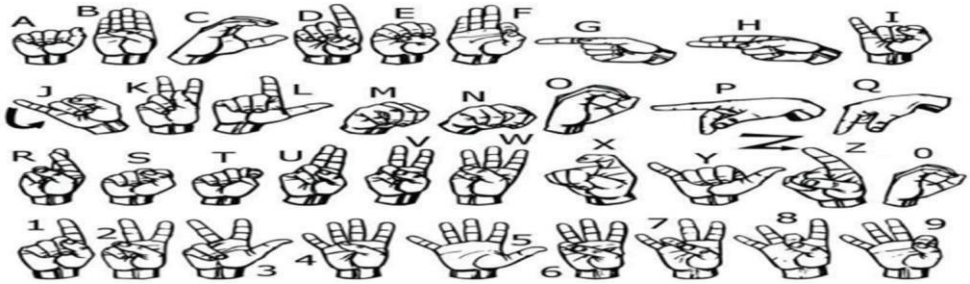


Fig. 1 ASL Hand gestures

II. THEORETICAL FRAMEWORK

A. Data Collection

In our research, initial attempts to find suitable datasets proved unsuccessful, as available datasets were not in the required raw image format but rather in RGB values. Consequently, we opted to create our own dataset using the OpenCV library. In order to generate the dataset, collect various images of hand signs for that here we use open computer vision (open cv) library in python to detect objects. First, we have to capture images for training and testing purpose we can capture those images through the webcam of the system[2]. The process involved capturing approximately 100 images for each of the ASL symbols A, B, and C, ensuring the dataset met our specific needs.

Fig. 2 Sign Recognition

B. Data Preprocessing

Video to Image Conversion is a pivotal step in sign language recognition, involving the extraction of frames from videos to produce still images, which are indispensable for training recognition models. Image quality is refined by denoising, contrast adjustment, and resizing, tailored to the specific needs of the dataset we captured around 800 images of each of the symbol in ASL for traini ng purposes and around 200 images per symbol for testing purpose. First, we capture each frame shown by the webcam of our machin e. In each frame we define a region of interest (ROI) which is denoted by a blue bounded square as shown in the image below. From the whole image we extracted our ROI which is RGB and convert it into grey scale Image. Finally, we apply our gaussian blur filter to our image which helps us extracting various features of our image[17]. Features extracted from the images are meticulously labeled with corresponding sign language glosses or spoken language translations, serving as target outputs for training and evaluating model accuracy. Apply Gaussian Blur filter and threshold to the frame taken with open cv to get the processed image after feature extraction[2] Accurate hand detection and landmark identification precede dataset augmentation, where techniques like rotation, translation, and flipping diversify hand poses. Extracted features are labeled with sign language glosses or translations, aiding supervised learning. These labeled datasets form the basis for training and evaluating sign language recognition models, ensuring reliable communication accessibility for users, and highlighting the importance of video-to-image conversion in system development. These meticulous processes culminate in a successful video-to-image transformation, ultimately enhancing accessibility and facilitating improved communication for users worldwide.

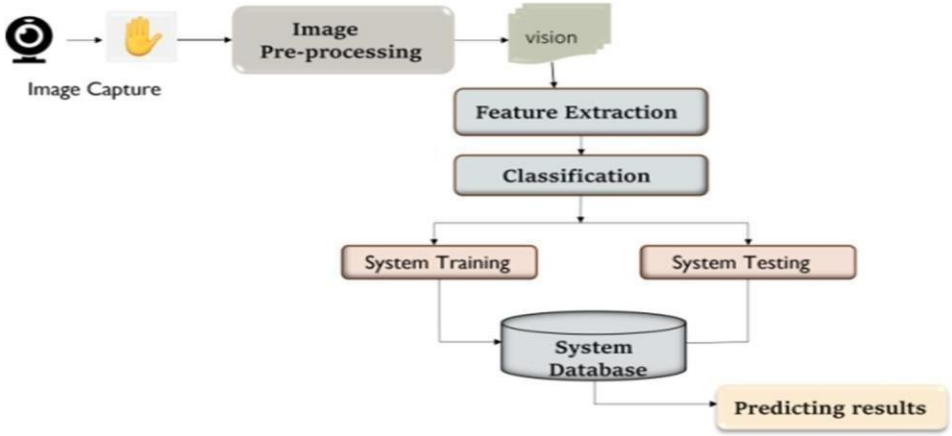


Fig. 3 Process flow

C. Train Model

Then we build a random forest classifier for hand gesture recognition task. It loads previously extracted hand landmark data, splits it into training and testing sets, trains a random forest classifier, evaluates its accuracy and saves the trained model to a file. Pickle module is used to load and save python objects as serialized binary data. Scikit-learn is used for building and evaluating the random forest classifier, as well as for splitting the dataset. Numpy is used for numerical operations. Split the data into training and testing sets using scikit-learns train_test_split function. Here 80% of the data is used for training (x_train, y_train) and 20% is used for testing (x_test, y_test) the stratify parameter ensures that the class distribution is preserved in the split. The user shows a hand gesture of any word based on the gesture language the data which is being provided in the model and also used for the testing the model [2]. Serialize and save the trained random forest model to a file called 'model.p' using pickle. This allows to use the trained model for inference without retraining it.

D. Testing

Utilizing a pretrained model, a neural network previously trained on extensive datasets, we classify hand gestures captured in real-time via a webcam using the MediaPipe library. We convert our input images (RGB) into grayscale and apply gaussian blur to remove unnecessary noise. We applied adaptive threshold to extract our hand from the background and resize our images to 128 x 128. We feed the input images after preprocessing to our model for training and testing after applying all the operations [17]. Our model actively captures frames from the webcam feed, effectively pinpointing hand landmarks through advanced algorithms integrated within the MediaPipe framework. Subsequently, based on the detected landmarks, the model makes predictions regarding the specific gestures being performed. When the user shows a gesture of any specific letter, then the model recognizes the gesture [2]. This prediction process occurs instantaneously, allowing for swift and accurate recognition of gestures as they unfold in real-time. The culmination of this process is the immediate display of the recognized gestures, providing users with timely and actionable feedback regarding their hand movements.

III. LITERATURE REVIEW

1) Paper I: Conversion of Sign Language into Text Using Machine Learning Technique [8]

This study introduces a novel approach to address communication barriers encountered by individuals who are deaf or dumb, focusing on converting hand gestures into text utilizing Convolutional Neural Networks (CNN). Given the fundamental importance of communication, particularly for those with hearing impairments, sign language serves as a vital means of expression. However, the absence of interpreters often hinders effective communication. The proposed methodology involves capturing hand gestures via camera using OpenCV and processing them through a CNN model, which includes convolution and pooling layers for feature extraction, followed by classification through fully connected layers and a Softmax output layer. The incorporation of a diverse dataset enhances the accuracy of the model. This innovative system aims to improve communication and foster inclusivity for impaired individuals by harnessing machine learning techniques and advanced image processing methodologies.

2) Paper II: Conversion of Gesture Language to Text Using OpenCV and CNN[3]

This study proposes an intuitive application to bridge communication gaps for deaf and mute individuals by translating hand gestures into text efficiently. Through the utilization of convolutional neural networks and machine learning algorithms, the system captures gestures via webcam and converts them into alphabets and words, fostering easy comprehension. A user-friendly interface enhances accessibility, featuring automatic spelling correction facilitated by Python libraries. By customizing a dataset with OpenCV to gather hand sign images and applying Gaussian blur for feature extraction, the system ensures robust gesture classification. Training the TensorFlow model involves preprocessing input images and testing for accuracy, with subsequent finger spelling implementation refining letter prediction based on gesture counts. With a final accuracy of 98.0% in real-time American Sign Language recognition, this innovative approach promises to revolutionize communication accessibility for the deaf and mu te community globally.

IV. METHODOLOGY

Our application addresses system constraints by capturing webcam gestures and translating them into text using convolutional neural networks and machine learning algorithms. Through a user-friendly interface, it incorporates auto-correction functionalities via Python libraries for accurate gesture-to-text conversion. This comprehensive approach ensures seamless communication, enabling individuals unfamiliar with sign language to understand gestures effectively.

This comprehensive approach promises to enhance usability and efficiency in overcoming existing constraints. TensorFlow has an inbuilt function to calculate the cross entropy [9]. Additionally, the inclusion of word spell correction functionality enhances the precision of the model, rectifying potential errors and further refining the clarity of the conveyed message. By offering advanced features and robust functionality, our model enhances the communication experience for all users. The system is a vision-based approach. All the signs are represented with bare hands and so it eliminates the problem of using any artificial devices for interaction [17]. CNNs are inspired by the visual cortex of the human brain. The artificial neurons in a CNN will connect to a local region of the visual field, called a receptive field. This is accomplished by performing discrete convolutions on the image with filter values as trainable weights. Multiple filters are applied for each channel, and together with the activation functions of the neurons, they form feature maps. This is followed by a pooling scheme, where only the interesting information of the feature maps are pooled together [6].

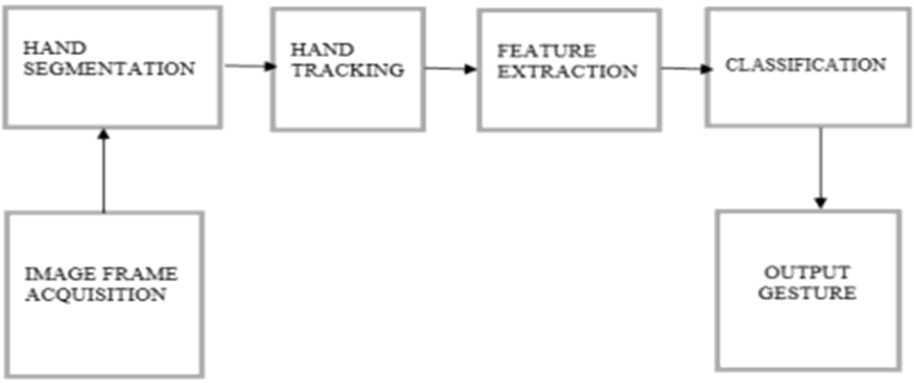


Fig. 4 Gesture detection process

Through these comprehensive features, the model contributes significantly to bridging communication gaps and facilitating seamless interaction for users with hearing impairments. PCA is a technique that allow to represent pictures as points during a low-dimensional space. If every image consists of 32x32 pixels whose values vary from zero to 255, then every image defines some points in 1024-dimensional space. If one tends to grab a sequence of pictures representing a gesture then this sequence can generate a sequence of points in space However, this set of points can sometimes lie on a low-dimensional sub-space inside the world 1024D space. The PCA algorithmic rule permits us to search out this sub space that sometimes consists of up to three dimensions. Th is enables us to examine the sequence of points representing the gesture [1].

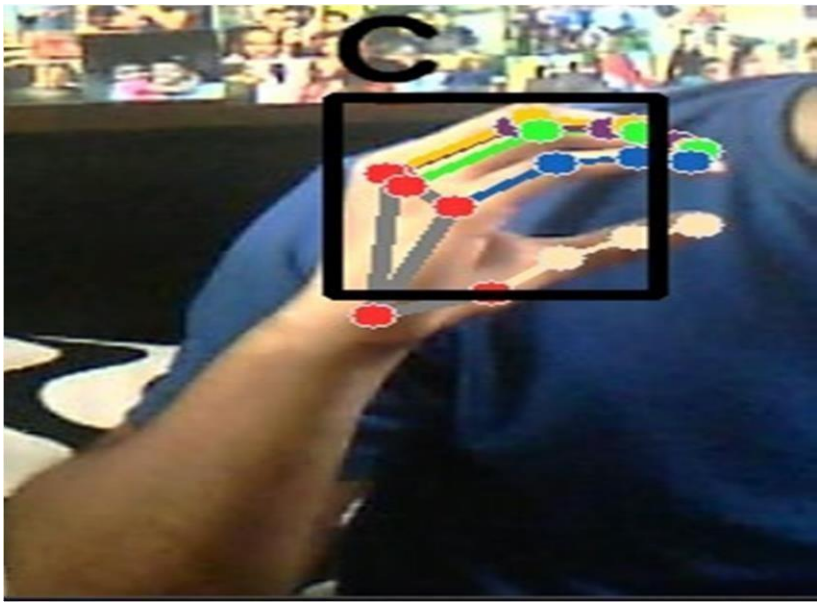


Fig. 5 Recognizing Alphabets

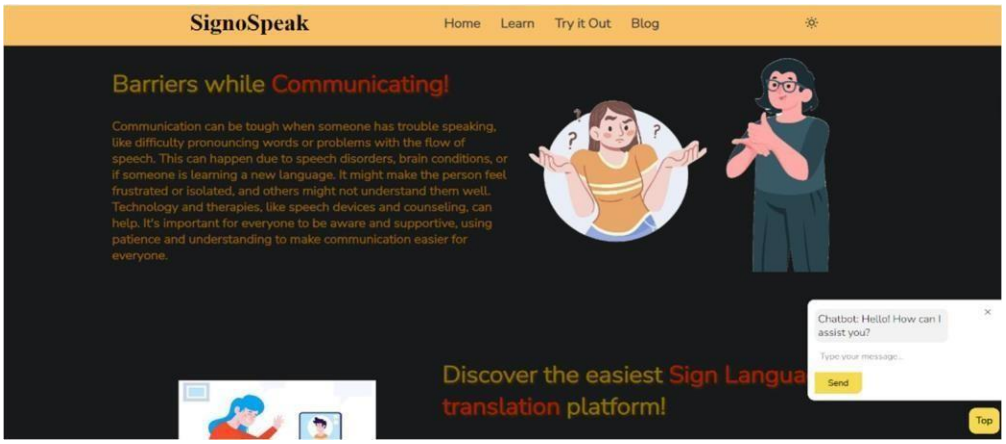


Fig. 6 SignoSpeak website

V. RESULT

In this research, a functional real time vision based American sign language recognition for Deaf and Dumb people have been developed for asl alphabets. We achieved final accuracy of 95.7% on our dataset. We are able to improve our prediction after implementing two layers of algorithms in which we verify and predict symbols which are more similar to each other. This way we are able to detect almost all the symbols provided that they are shown properly, there is no noise in the background and lighting is adequate.

VI. CONCLUSION

In our ongoing endeavour to enhance our sign language to text conversion system, we are dedicated to achieving unparalleled accuracy and precision through continuous refinement and optimization. We are expanding our dataset comprehensively to encompass a wide range of gestures, expressions, and languages, ensuring the robustness and adaptability of our system. Embracing a global perspective, we are integrating additional sign languages to promote inclusivity and accessibility across diverse linguistic and cultural backgrounds. Our focus also extends to developing a user-friendly interface that prioritizes accessibility, facilitating seamless interaction for users. As we progress, we remain committed to leveraging cutting-edge technologies to perfect our model, with the ultimate goal of empowering the global sign language community. Through our efforts, we aim to make a significant impact on individuals who rely on sign language as their primary means of communication, fostering more effective and expressive interactions worldwide. Our model provides 95.7 % accuracy for the 26 letters of the alphabet and 10 numeric digits.

VII. ACKNOWLEDGMENT

We are profoundly grateful to Dr. Varsha Shah, Principal of Rizvi College of Engineering, Mumbai for her expert guidance and continuous encouragement throughout to see that this research rights its target.

We would like to express deepest appreciation towards Dr. Anupam Choudhary, Head of Department and Prof. Shiburaj Pappu, Dean of Computer department whose invaluable guidance supported us in this research.

At last, we would express our sincere heartfelt gratitude to all the staff members of computer engineering department who helped us directly or indirectly during this course of work.

REFERENCES

[1] T. Yang, Y. Xu, and "A., Hidden Markov Model for Gesture Recognition", CMURI-TR-94 10, Robotics Institute, Carnegie Mellon Univ., Pittsburgh, PA, May 1994.

[2] Pujan Ziaie, Thomas M uller, Mary Ellen Foster, and Alois Knoll "A Na¨ive Bayes Munich, Dept. of Informatics VI, Robotics and Embedded Systems, Boltzmannstr. 3, DE-85748 Garching, Germany.

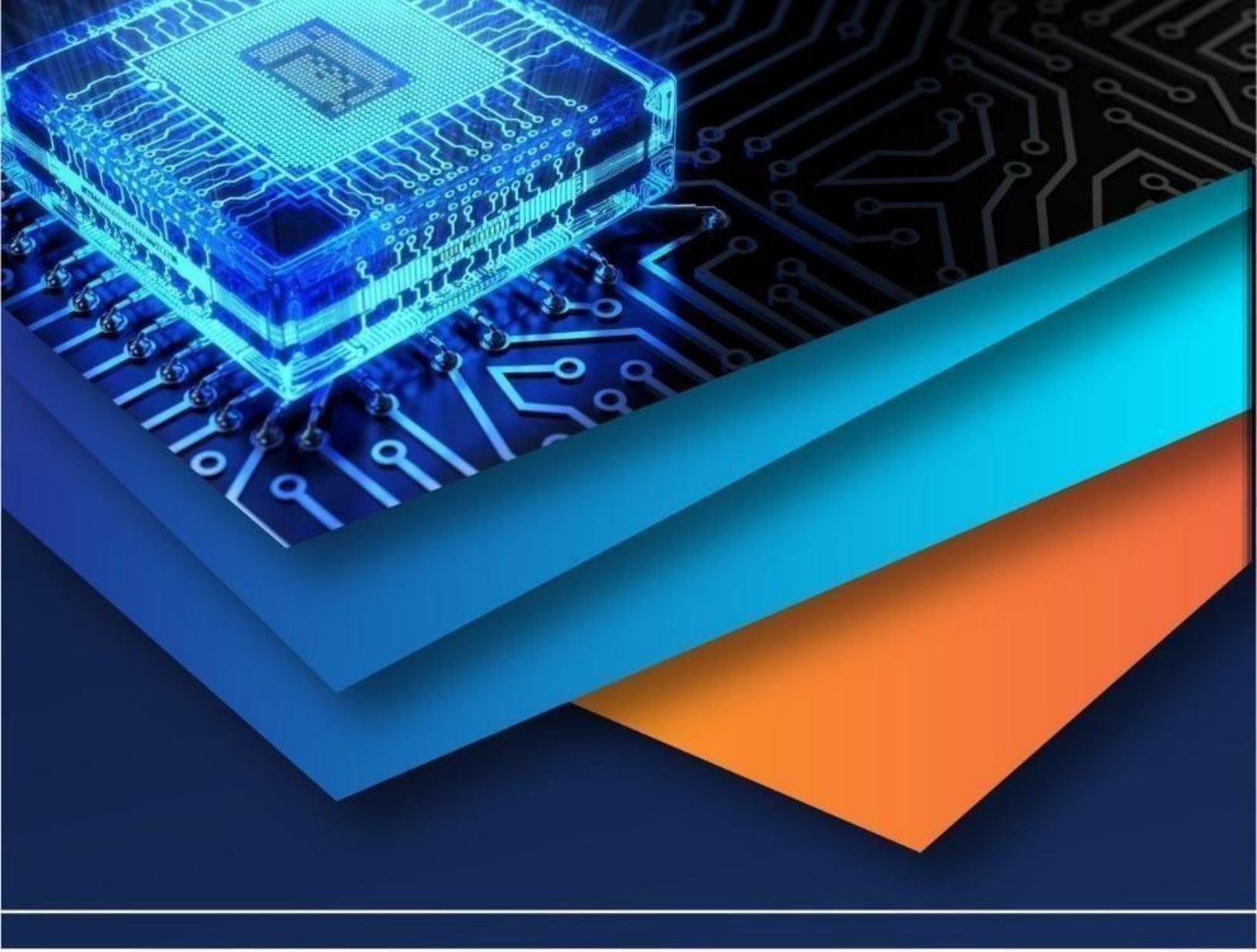
[3] Mohammed Waleed Kalous, Machine recognition of Auslan signs using PowerGloves: Towards large-lexicon recognition of sign language.

[4] Byeongkeun Kang, Subarna Tripathi, Truong Q. Nguyen" Real-time sign language fingerspelling recognition using convolutional neural networks from depth map" 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)

[5] ijraset.com/best-journal/sign-language-interpreter

[6] Pigou L., Dieleman S., Kindermans PJ., Schrauwen B. (2015) Sign Language Recognition Using Convolutional Neural Networks. In: Agapito L., Bronstein M., Rother C. (eds) Computer Vision - ECCV 2014 Workshops. ECCV 2014. Lecture Notes in Computer Science, vol 8925. Springer, Cham

[7] Zaki, M.M., Shaheen, S.I.: Sign language recognition using a combination of new vision-based features. Pattern Recognition Letters 32(4), 572–577 (2011)
[8] ijraset.com/best-journal/conversion-of-sign-language-video-to-text-and-speech
[9] N. Mukai, N. Harada and Y. Chang, "Japanese Fingerspelling Recognition Based on Classification Tree and Machine Learning," 2017 Nicograph International (NicoInt), Kyoto, Japan, 2017, pp. 19-24. doi:10.1109/NICOInt.2017.9
[10] Number System Recognition (https://github.com/chasinginfinity/number-sign-recognition)
[11] Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., Bengio, Y.: Theano: a CPU and GPU math expression compiler. In: Proceedings of the Python for Scientific Computing Conference (SciPy), June 2010, oral Presentation
[12] "Sign language recognition." In visual Analysis of Humans, pp. 539- 562. Springer London,2011. Cooper, Helen, Brian Holt, and Richard Bowden.
[13] "Handshape recognition for Argentinian sign language using problem". Journal of Computer Science and Technology 16(2016). Ronchetti, Franco, Facundo Quiroga, Cesar Armando Estrebou and Laura Cristina Lanzarini.
[14] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 20(12):1371–1375, Dec 1998
[15] S. Liwicki and M. Everingham. Automatic recognition of fingerspelled words in british sign language. In Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on, pages 50–57, June 2009.
[16] J. Isaacs and S. Foo. Hand pose estimation for american sign language recognition. In System Theory, 2004. Proceedings of the Thirty-Sixth Southeastern Symposium on, pages 132–136, 2004
[17] https://www.ijert.org/a-review-paper-on-sign-language-recognition-for-the-deaf-and-dumb



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY


ISSN No. : 2321-9653
iJRASET
International Journal for Research in Applied Science & Engineering Technology
IJRASET is indexed with Crossref for DOI-DOI : 10.22214
Website : www.ijraset.com, E-mail : ijraset@gmail.com
ISRA Journal Impact Factor: 7.429
45.98
INDEX COPERNICUS
THOMSON REUTERS
Researcher ID: N-9681-2016
10.22214/IJRASET
TOGETHER WE REACH THE GOAL
SJIF 7.429
Certificate
It is here by certified that the paper ID : IJRASET59787, entitled
SignoSpeak: Bridging the Gap
by
Utsav G. Kuntalwad
after review is found suitable and has been published in
Volume 12, Issue IV, April 2024
in
International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors
Editor in Chief, iJRASET


ISSN No. : 2321-9653
iJRASET
International Journal for Research in Applied Science & Engineering Technology
IJRASET is indexed with Crossref for DOI-DOI : 10.22214
Website : www.ijraset.com, E-mail : ijraset@gmail.com
ISRA Journal Impact Factor: 7.429
45.98
INDEX COPERNICUS
THOMSON REUTERS
Researcher ID: N-9681-2016
10.22214/IJRASET
TOGETHER WE REACH THE GOAL
SJIF 7.429
Certificate
It is here by certified that the paper ID : IJRASET59787, entitled
SignoSpeak: Bridging the Gap
by
Srushti S. Sawant
after review is found suitable and has been published in
Volume 12, Issue IV, April 2024
in
International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors
Editor in Chief, iJRASET


ISSN No. : 2321-9653
iJRASET
International Journal for Research in Applied Science & Engineering Technology
IJRASET is indexed with Crossref for DOI-DOI : 10.22214
Website : www.ijraset.com, E-mail : ijraset@gmail.com
ISRA Journal Impact Factor: 7.429
45.98
INDEX COPERNICUS
THOMSON REUTERS
Researcher ID: N-9681-2016
10.22214/IJRASET
TOGETHER WE REACH THE GOAL
SJIF 7.429
Certificate
It is here by certified that the paper ID : IJRASET59787, entitled
SignoSpeak: Bridging the Gap
by
Mayur R. Kyatham
after review is found suitable and has been published in
Volume 12, Issue IV, April 2024
in
International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors
Editor in Chief, iJRASET


ISSN No. : 2321-9653
iJRASET
International Journal for Research in Applied Science & Engineering Technology
IJRASET is indexed with Crossref for DOI-DOI : 10.22214
Website : www.ijraset.com, E-mail : ijraset@gmail.com
ISRA Journal Impact Factor: 7.429
45.98
INDEX COPERNICUS
THOMSON REUTERS
Researcher ID: N-9681-2016
10.22214/IJRASET
TOGETHER WE REACH THE GOAL
SJIF 7.429
Certificate
It is here by certified that the paper ID : IJRASET59787, entitled
SignoSpeak: Bridging the Gap
by
Prerna S. Shakwar
after review is found suitable and has been published in
Volume 12, Issue IV, April 2024
in
International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors
Editor in Chief, iJRASET


ISSN No. : 2321-9653
iJRASET
International Journal for Research in Applied Science & Engineering Technology
IJRASET is indexed with Crossref for DOI-DOI : 10.22214
Website : www.ijraset.com, E-mail : ijraset@gmail.com
ISRA Journal Impact Factor: 7.429
45.98
INDEX COPERNICUS
THOMSON REUTERS
Researcher ID: N-9681-2016
10.22214/IJRASET
TOGETHER WE REACH THE GOAL
SJIF 7.429
Certificate
It is here by certified that the paper ID : IJRASET59787, entitled
SignoSpeak: Bridging the Gap
by
Dr. Varsha Shah
after review is found suitable and has been published in
Volume 12, Issue IV, April 2024
in
International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors
Editor in Chief, iJRASET