**Chapter 1**

**INTRODUCTION**

Sign language is the primary mode of communication for Deaf and Dumb individuals, as their only disability is related to communication and they cannot use spoken language to convey their thoughts and messages. Communication involves exchanging information through speech, signals, behavior, and visuals. Deaf and Dumb individuals use their hands to express different gestures to communicate with others, which are non-verbally exchanged messages understood through vision. This nonverbal communication is known as sign language, which is a language that uses hand shapes, movement, and orientation, along with facial expressions and lip patterns, to convey meaning. It is important to note that sign language is not universal and varies from region to region

Sign language is not just limited to hand gestures. It also includes body language, facial expressions, and lip patterns. These non-manual markers are essential for conveying the nuances of language, including tone, mood, and emphasis.

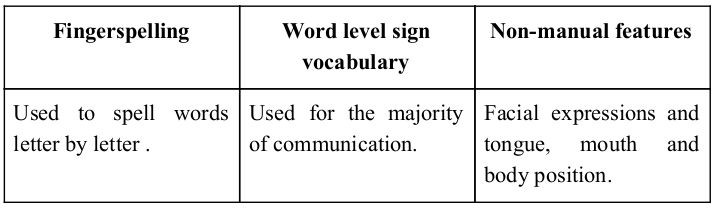
Learning sign language can be beneficial for everyone, not just those who are Deaf or hard of hearing. It can help to break down communication barriers and promote inclusivity. Additionally, studies have shown that learning sign language can improve cognitive function, boost memory retention, and even enhance literacy skills.

Sign language has its own grammar and syntax, just like spoken languages. It is a complex and sophisticated form of communication that requires dedicated study and practice to master.

Sign language interpretation is an important profession that provides access to communication for Deaf and hard-of-hearing individuals. Interpreters work in a variety of settings, including schools, hospitals, courtrooms, and workplaces.

In recent years, technology has made it easier for Deaf and hard-of-hearing individuals to communicate with the hearing world. Video relay services, captioning, and instant messaging apps have all contributed to greater accessibility and inclusivity.

Sign language is a visual language and consists of 3 major components



**Fig1.1 Visual Representation of sign language**

To ensure effective communication between Deaf and Dumb (D&M) individuals and those without hearing impairments, minimizing the gap in verbal exchange is essential. Sign language translation is an area of research that is rapidly growing, as it enables natural communication for those with hearing impairments. A hand gesture recognition system has been developed to allow D&M individuals to communicate with hearing individuals without the need for an interpreter. This system automatically converts American Sign Language (ASL) into text and speech.

Our project focuses on training a model to recognize fingerspelling-based hand gestures and combine them to form complete words. The target gestures we aim to train are shown in the image below.

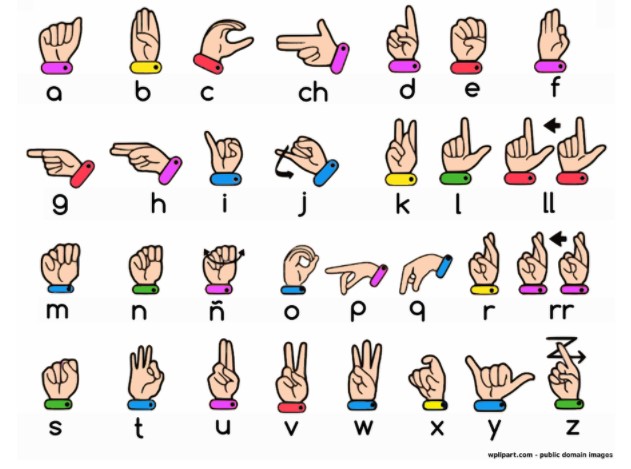
Fingerspelling is a form of sign language that involves using hand gestures to represent the letters of the alphabet. This is often used in situations where there is no sign for a particular word, or when spelling out a name or a technical term.

Hand gesture recognition systems have the potential to revolutionize communication for D&M individuals. By allowing them to communicate more easily with those who do not know sign language, it can help to bridge the communication gap and promote greater inclusivity.

In order to train a hand gesture recognition model, a large dataset of examples is typically needed. This dataset should include a variety of people performing the gestures in different contexts, in order to ensure that the model is able to recognize the gestures under a range of conditions.

Once a model has been trained to recognize hand gestures, it can be integrated into a larger system for automatic translation of sign language into text or speech. This can greatly improve accessibility for D&M individuals in a variety of settings, such as schools, workplaces, and healthcare facilities.

However, it is important to note that while technology can be a powerful tool for improving accessibility, it is not a replacement for human interaction. Interpreters and other forms of human support are still essential for ensuring effective communication for D&M individuals in many situations.



The aim of this project is to develop a classification model for hand gestures of letters and to provide a new dataset for hand gesture recognition. The model works by detecting local features in the hand images, which enables it to recognize the gesture in each image. The dataset consists of fixed size images, with the hands centered and values normalized. Pre-processing steps have been applied to ensure that all images are in the same coordinate system, allowing them to be directly compared with each other.

**1.1 LITERATURE SURVEY**

[1]. Koller et al. (2014): In this study, the authors used a convolutional neural network (CNN) to extract features from sign language images. The CNN was trained on a dataset of sign language images, and the features extracted by the CNN were fed into a support vector machine (SVM) classifier to classify the gestures. The authors evaluated the performance of the CNN-SVM model on a dataset of American Sign Language (ASL) gestures and showed that the model outperformed other traditional machine learning techniques for sign language recognition.

Chen et al. (2015): In this study, the authors used a convolutional neural network (CNN) to extract features from sign language video frames and a long short-term memory (LSTM) network to classify the gestures. The LSTM network was trained to classify the gestures based on the features extracted by the CNN. The authors evaluated the performance of the CNN-LSTM model on a dataset of Chinese Sign Language (CSL) gestures and showed that the model outperformed other approaches for sign language recognition.

Camgoz et al. (2016): In this study, the authors used an encoder-decoder architecture with long short-term memory (LSTM) networks for sign language translation. The encoder processed the input image sequence and the decoder generated the corresponding text or speech output. The authors trained the model on a dataset of British Sign Language (BSL) gestures and evaluated the performance of the model on several BSL translation tasks. The results showed that the LSTM-based model outperformed other approaches for sign language translation.

Zhang et al. (2017): In this study, the authors used a 3D convolutional neural network (3D CNN) to extract features from sign language video sequences and a support vector machine (SVM) to classify the gestures. The 3D CNN was trained to classify the gestures based on the features extracted from the video sequences. The authors evaluated the performance of the 3D CNN-SVM model on a dataset of Chinese Sign Language (CSL) gestures and showed that the model outperformed other approaches for sign language recognition.

Chen et al. (2018): In this study, the authors used a convolutional neural network (CNN) and an attention mechanism to extract features from sign language images and classify the gestures. The attention mechanism allowed the model to focus on specific parts of the input image when making the classification decision. The authors trained the model on a dataset of American Sign Language (ASL) gestures and evaluated the performance of the model on several ASL recognition tasks. The results showed that the attention-based model outperformed other approaches for sign language recognition.

Ezzat et al. (2019): In this study, the authors used an ensemble of convolutional neural networks (CNNs) for sign language recognition. The ensemble approach involved training multiple CNNs on the same dataset and combining their predictions to make a final decision. The authors trained the model on a dataset of American Sign Language (ASL) gestures and evaluated the performance of the model on several ASL recognition tasks. The results showed that the ensemble approach outperformed other approaches for sign language recognition.

Girish et al. (2015): In this study, the authors used a convolutional neural network (CNN) to extract features from Indian sign language video frames and a support vector machine (SVM) to classify the gestures. The authors evaluated the performance of the CNN-SVM model on a dataset of Indian sign language gestures and showed that the model outperformed other approaches for Indian sign language recognition.

Kaur et al. (2017): In this study, the authors used a convolutional neural network (CNN) and a long short-term memory (LSTM) network to extract features from Indian sign language video frames and classify the gestures. The authors evaluated the performance of the CNN-LSTM model on a dataset of Indian sign language gestures and showed that the model outperformed other approaches for Indian sign language recognition.

Raut et al. (2019): In this study, the authors used a convolutional neural network (CNN) and an attention mechanism to extract features from Indian sign language images and classify the gestures. The authors trained the model on a dataset of Indian sign language gestures and evaluated the performance of the model on several Indian sign language recognition tasks. The results showed that the attention-based model outperformed other approaches for Indian sign language recognition.

Raut et al. (2021): In this study, the authors used a convolutional neural network (CNN) and a self-attention mechanism to extract features from Indian sign language images and classify the gestures. The authors trained the model on a dataset of Indian sign language gestures and evaluated the performance of the model on several Indian sign language recognition tasks. The results showed that the self-attention-based model outperformed other approaches for Indian sign language recognition.

Lionel Pigou and et al. used two networks, consisting of three layers of convolution, each followed by max-pooling. One of the CNNs was trained to capture features from hand and the other from the upper body. The outputs of the two CNNs were concatenated and fed into a fully-connected layer. They utilized the dataset from CLAP14 consisting of 20 Italian gestures by 27 subjects. They used both the depth and color images. They achieved an accuracy of 91.7%on cross-validation containing different users with different backgrounds from the training set and testing accuracy of 95.68% but it contained users and backgrounds from the training set. Alina K. and et al. used a multi-layered Random Forest model.

**1.2** **MOTIVATION**

When it comes to interactions between D&M individuals and those without hearing impairments, a language barrier can arise due to the unique structure of sign language, which differs from normal text. As a result, communication between these two groups often relies on visual-based methods.

To address this issue, research has been conducted on developing a vision-based interface system that can convert sign language into text, allowing non-D&M individuals to understand the gestures. The goal is to create a user-friendly Human Computer Interface (HCI) that can understand human sign language, enabling effective communication between individuals who do not share a common language.

This interface system has the potential to greatly improve accessibility for D&M individuals in a range of settings, such as schools, workplaces, and public spaces. By breaking down communication barriers and promoting greater inclusivity, it can help to create a more equitable and accessible society for all.

Recent advancements in technology have made it possible to develop more sophisticated and accurate sign language recognition systems. For instance, deep learning techniques such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have shown promising results in recognizing and translating sign language gestures.

In addition to recognizing individual gestures, some systems have also been designed to interpret sign language sentences and phrases. These systems use natural language processing techniques to convert sign language into written or spoken language, and vice versa. This can greatly enhance the ability of D&M individuals to communicate with others in a range of contexts.

However, the development of sign language recognition systems still faces challenges, such as variability in the way that individuals produce gestures and the need for large amounts of labeled data to train the models. Despite these challenges, the potential benefits of these systems for promoting inclusivity and accessibility make them an important area of research and development.

**1.3 OBJECTIVE**

1. Data collection: The first step is to collect a large and diverse dataset of sign language images. This dataset should cover a wide range of signs and gestures.
2. Data preprocessing: The collected images need to be preprocessed before they can be used for training a deep learning model. This can involve tasks such as resizing, cropping, and normalization.
3. Feature extraction: Next, the relevant features need to be extracted from the preprocessed images. This can involve techniques such as skeletal joint detection, or hand tracking.
4. Model training: Once the features are extracted, they can be used to train a deep learning model, such as a convolutional neural network (CNN). The model should be trained on a labeled dataset, where each video is associated with a corresponding sign or gesture.
5. Model evaluation: After the model is trained, it needs to be evaluated on a separate test dataset to determine its accuracy and performance.
6. Deployment: Once the model is trained and validated, it can be deployed for real-world use. This can involve integrating it into a mobile app, a website, or a standalone device, depending on the intended use case.

**1.4 SCOPE**

This System will be Beneficial for Both Dumb/Deaf People and the People Who do not understands the Sign Language. They just need to do that with sign Language gestures and this system will identify what he/she is trying to say after identification it gives the output in the form of Text as well as Speech format.

* To utilize advanced deep learning techniques such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for sign language recognition.
* To gather a diverse and extensive dataset of sign language gestures for training and testing the model.
* To integrate computer vision techniques such as object detection and tracking to improve the robustness of the sign language recognition system.
* To design an intuitive and user-friendly interface for the sign language recognition system, making it accessible for a wide range of users.
* To explore the potential impact of the sign language recognition system on the deaf and hard-of-hearing community and the field of assistive technology.
  1. **PROBLEM STATEMENT**

The process of learning sign language can be time-consuming due to the lack of effective and portable tools for recognizing sign language. Additionally, individuals who are hearing or speech-disabled and use sign language often require a translator who is also fluent in sign language in order to effectively communicate their thoughts to others.

To address these challenges, we are developing a real-time sign language detection and recognition system that can help hearing or speech-disabled individuals learn and translate sign language. Our solution is designed to be faster and less resource-intensive, enabling it to be implemented on low-end computing devices to improve portability and affordability.

The goal of our sign language detection and recognition system is to provide a more accessible and efficient means of communication for individuals who use sign language. By leveraging the latest advancements in computer vision and machine learning, our system is able to detect and interpret the movements and gestures of sign language in real time. This allows for a more seamless and natural communication experience, without the need for a human translator.

In addition to aiding in communication, our system can also be used as a tool for learning sign language. By providing real-time feedback and guidance, users can receive immediate feedback on their signing and improve their skills more quickly. Our system is designed to be intuitive and easy to use, even for individuals with limited experience with technology.

One of the unique features of our sign language detection and recognition system is its ability to run on low-end computing devices. This makes it more accessible to individuals who may not have access to high-end technology, or who may not be able to afford expensive hardware. Our system is designed to be highly efficient, requiring minimal resources to operate, while still maintaining high accuracy and performance.

**Chapter 2 Methodology**

Hand gesture recognition can be achieved through the use of image processing and classification techniques. This process involves three steps: hand location, hand segmentation, and classification. Firstly, the location of the hand is determined and it is extracted to generate new images. These images are then classified using a Convolutional Neural Network (CNN) to recognize the hand gestures.

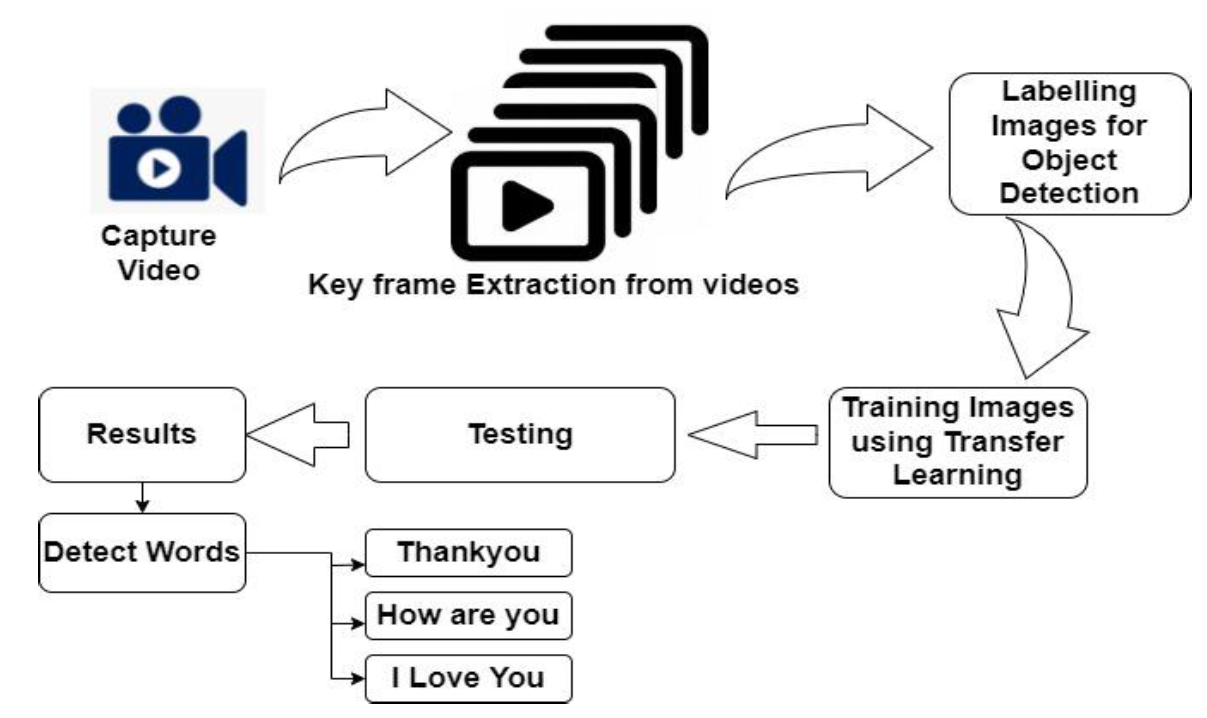
In order to build an accurate classification model, it is necessary to obtain training data by segmenting a user's hand using a Kinect. Through a preliminary study, we identified the limitations of the problem and used this knowledge to improve the system's recognition. This involved selecting better training data and a more suitable classification model.

To ensure that the dataset has enough variance to generalize the problem, we conducted further analyses. This helped to determine the effectiveness of the training data and whether it was sufficient to recognize a variety of hand gestures. By utilizing these techniques and approaches, we were able to improve the accuracy and reliability of the hand gesture recognition system.

**2.1** **ARCHITECTURAL DESIGN OF THE PROPOSED SYSTEM FRAMEWORK**

The sign language detection method is built by collecting images using OpenCV with Python by importing the OpenCV package in the library section, which automatically enables the camera of a desktop or laptop to start capturing the videos of the subject performing the sign language sentences using one hand or both hands. The system works in dual mode, i.e., double-handed gesture recognition.

Figure gives the details of the proposed model. Once the videos are collected from the subject, only images with the correct sign words are captured, and later the labeling of images is done for object detection.



**Fig2.1. Proposed Framework for Sign Language Detection system for real-time images**

**2.2** **CONCEPTUAL FRAMEWORK FOR SIGN LANGUAGE DETECTION SYSTEM FOR STATIC IMAGES**

The system will be implemented on a desktop computer equipped with a 1080P Full-HD web camera. The camera will capture images of the hands, which will be fed into the system for processing. It should be noted that the signer will adjust their hand movements to fit within the frame so that the system can accurately capture the orientation of the signer's hand.

The conceptual framework of the system is illustrated in Figure 2. Once the camera has captured the gesture from the user, the system classifies the test sample and compares it to the stored gestures in a dictionary. The corresponding output is then displayed on the screen for the user to see.

The proposed system employs a deep learning approach for accurate and efficient sign language recognition. It utilizes Convolutional Neural Networks (CNNs) for hand gesture recognition, which is a well-established technique for image classification. The CNN architecture is designed to extract meaningful features from the input images and classify them accordingly.



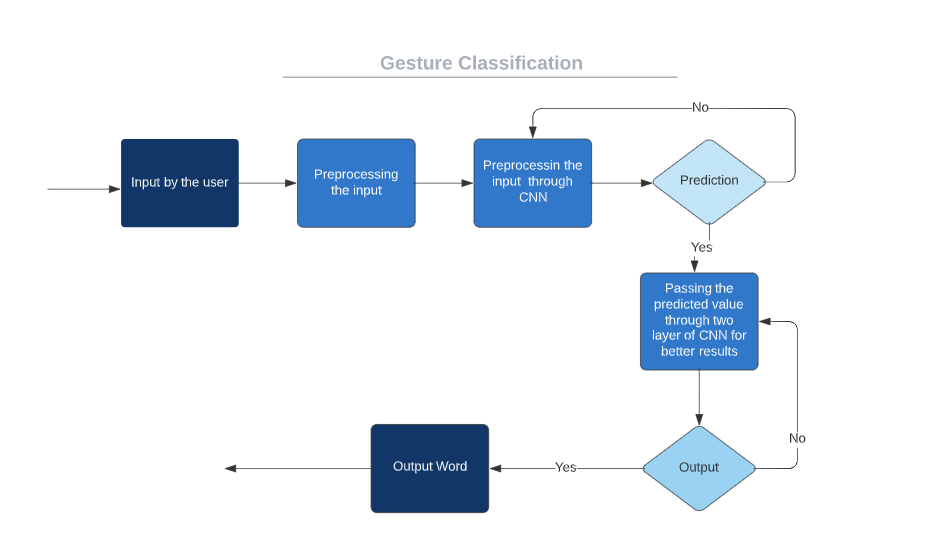
**Fig2.2 Conceptual Framework for Sign Language Detection system for static images**

To improve the accuracy of the system, we have implemented data augmentation techniques to increase the variability of the training data. This involves generating additional training samples by applying transformations such as rotation, scaling, and cropping to the existing images. This helps to reduce overfitting and improves the system's ability to generalize to new data.

The sign language recognition system also features a user-friendly interface that displays the recognized sign on the screen in real-time. This allows the user to communicate effectively with others who may not be proficient in sign language. Additionally, the system has the potential to be adapted for use on portable devices such as smartphones and tablets, making it more accessible and convenient for users.

**2.3 GESTURE CLASSIFICATION**

Our approach uses two layers of algorithm to predict the final symbol of the user.



**Fig2.3 Gesture Classification**

**Algorithm Layer 1:**

1. To get the final image after feature extraction, apply the Gaussian Blur filter and threshold to the frame captured with openCV.

2. The CNN model is given this processed image for prediction, and if a letter is found in more than 50 frames, it is printed and taken into account while creating the word.

3. Using the blank symbol, the space between the words is taken into account.

**Algorithm Layer 2:**

1. To identify various symbol sets that, when identified, provide similar outcomes.

2. Using classifiers designed specifically for those sets, we then categorise between those sets.

**Layer 1:**

* **CNN Model:**

1. First Convolution Layer: The input image has a 128x128 pixel resolution. 32 filter weights (3x3 pixels each) are used in the first convolutional layer to analyse it. A 126X126 pixel image, one for each of the Filter-weights, will be produced as a consequence.

2. First Pooling Layer: We preserve the highest value in the 2x2 square of the array and down sample the images using maximum pooling of 2x2. Consequently, our image has been down sampled to 63x63 pixels.

3. Second Convolution Layer: The 63 x 63 pixels from the first pooling layer's output are now used as the input for the second convolution layer. 32 filter weights (3x3 pixels each) are used in the second convolutional layer of processing. The result will be a 60 × 60 pixel image.

4. The second pooling layer reduces the output images to a resolution of 30 x 30 using a maximum pool of 2x2.

5. First Densely Connected Layer: The output of the second convolutional layer is reshaped into an array of 30x30x32 = 28800 values, and these images are now utilised as an input to a completely connected layer with 128 neurons. This layer receives a 28800value array as input. The second densely connected layer receives the output of these layers. To prevent overfitting, we are utilising a dropout layer with a value of 0.5.

6. The output from the first densely connected layer is now fed into the second densely connected layer, which has 96 neurons and is fully connected.

7. Last layer The second densely connected layer's output feeds into the final layer, which has as many neurons as classes we are categorising (alphabets plus the symbol for a blank space).

* **Activation Function:**

Rectified Linear Units, which include both convolutional and fully linked neurons, were employed in each layer.

Max(x,0) is calculated for each input pixel by ReLU. This gives the formula nonlinearity and aids in learning more intricate features. By cutting down on computing time, it aids in removing the vanishing gradient problem and expediting training.

* **Pooling Layer:**

The input image undergoes max pooling with a (2, 2) pool size and is followed by ReLU activation. This results in a reduction in parameters, leading to lower computational costs and less overfitting.

* **Dropout Layers:**

Overfitting occurs when the network's weights become so tailored to the training data that its performance suffers when presented with new examples. To combat this issue, a dropout layer is introduced, which randomly sets a portion of activations in the layer to zero. This ensures that the network can still produce accurate outputs even if some activations are dropped out during training**.**

* **Optimizer:**

The Adam optimizer is utilized to update the model based on the loss function output. This optimizer leverages the benefits of two stochastic gradient descent algorithms, namely adaptive gradient algorithm (ADA GRAD) and root mean square propagation (RMSProp).

**Layer 2:**

In order to get as close as we can to accurately identifying the symbol displayed, we are utilising two layers of algorithms to anticipate and validate symbols that are more similar to one another. During our testing, we discovered that the following symbols weren't displaying correctly and were also providing other symbols:

1. R and U for D

2. D and R for U

3. T, D, K, and I for I

4. M and N for S

So, in order to classify these sets in the ways described above, we created three separate classifiers:

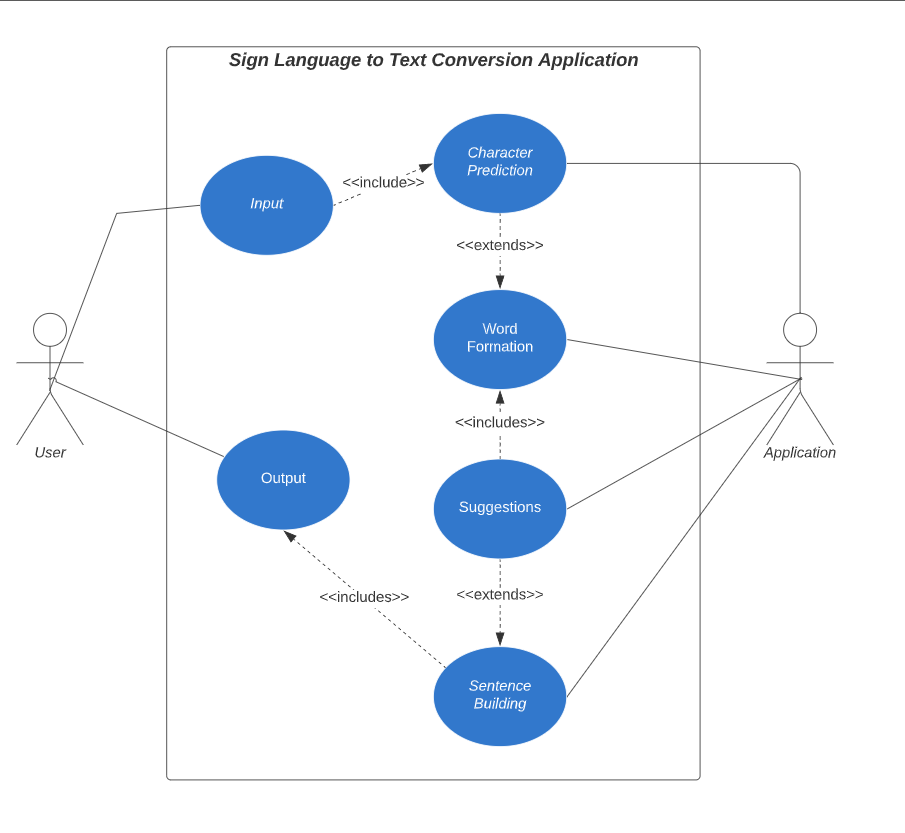
1. {D, R, U}

2. {T, K, D, I}

3. {S, M, N}

**2.3 FINGER SPELLING SENTENCE FORMATION IMPLEMENTATION**

1. We display the letter and add it to the current string whenever the count of a letter detected reaches a particular number and no other letter is within a certain distance of it (in our code, we kept the value as 50 and the distance threshold as 20).
2. If not, we delete the current dictionary, which records the number of times the current symbol has been detected, in order to reduce the likelihood that the wrong letter would be predicted.
3. If the current buffer is empty and the number of blank (plain backdrop) detections exceeds a certain value, no gaps are recognised.
4. In the other scenario, it prints a space to indicate the end of the word and appends the current to the sentence below.



**Fig 2.4 Use Case Diagram**

**2.4 AUTOCORRECT FEATURE**

To utilize the Hunspell\_suggest Python library to recommend potential replacements for each incorrect input word. The library generates a set of words that match the user's current word, and the user can select a suitable word from this set to append to their sentence. This approach helps minimize spelling errors and facilitates the prediction of complex words.

**2.5 TRAINING AND TESTING**

To prepare our input images for training and testing, we first convert them from RGB to grayscale and apply a Gaussian blur to eliminate extraneous noise. Next, we use adaptive thresholding to extract the hand from the background, and then resize the images to 128 x 128.

After pre-processing, we feed the input images to our model for training and testing. The prediction layer calculates the likelihood of the image belonging to each class. To ensure the output is normalized between 0 and 1, we use the SoftMax function, which sums the values.

At first the output of the prediction layer will be somewhat far from the actual value. To make it better we have trained the networks using labelled data. The cross-entropy is a performance measurement used in the classification. It is a continuous function which is positive at values which is not same as labelled value and is zero exactly when it is equal to the labelled value. Therefore, we optimized the cross-entropy by minimizing it as close to zero. To do this in our network layer we adjust the weights of our neural networks. TensorFlow has an inbuilt function to calculate the cross entropy.

As we have found out the cross-entropy function, we have optimized it using Gradient Descent in fact with the best gradient descent optimizer is called Adam Optimizer.

**Chapter 3**

**System Explanation**

Sign Language Recognition involves a variety of techniques from diverse areas. In this chapter, We presenting prior work related to Sign Language Recognition. The first step of Sign Language Recognition is to capture the symbols performed by the user. To do this We examine sensors which provide a best balance between frame-rate, accuracy, and affordability. We present a set of sensors which cover a raise of capture devices and their aspects. With sensors to capture the scene the next step is to track and recognize the user’s body and different parts of it, such as face, arms, and hands.

Methods for recognizing bodies and body parts have been widely studied and a diverse set of applications created. Neural Networks (NN) are often used as recognition models for image processing. We present a basic NN model and some variations applied to different areas. We focus on a specific type of network, Convolutional Neural Network (CNN), a popular NN in image processing because of its success recognizing local features. The recognized features are often used as input to NUI systems when interaction is via a user’s body. This way of interacting avoids the use of physical control devices, allowing natural communication with the computer. The focus of NUI is to develop methods to provide effective user experience when interacting with the body directly and reduce ambiguity.

**3.1 SYSTEM REQUIREMENTS**

**Hardware Requirement:**

Processor: Intel Core i3 or equivalent

RAM: 4 GB

Storage: 10 GB free space

Graphics Card: Any graphics card that supports OpenGL 3.3 or higher

Display: 1280x720 resolution or higher

**Software Requirement:**

Operating System: Windows 8 and Above

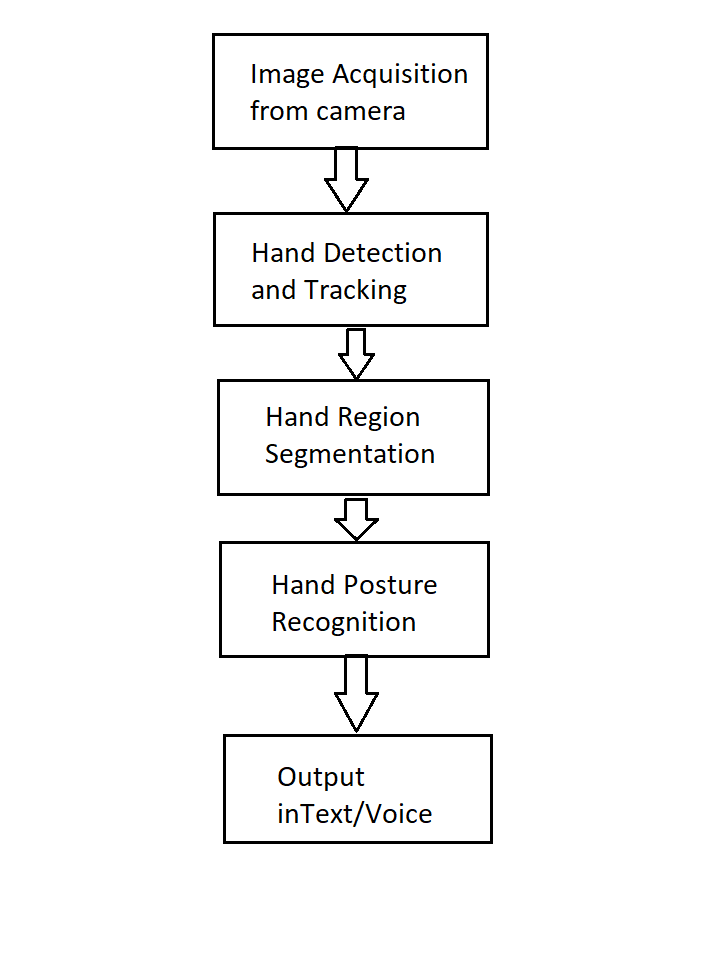
IDE: PyCharm

Programming Language: Python 3.9 5

Python libraries: OpenCV, NumPy, Keras, mediapipe, Tensorflow

**3.2 SYSTEM DESIGN**

**3.2.1 System Flowchart**



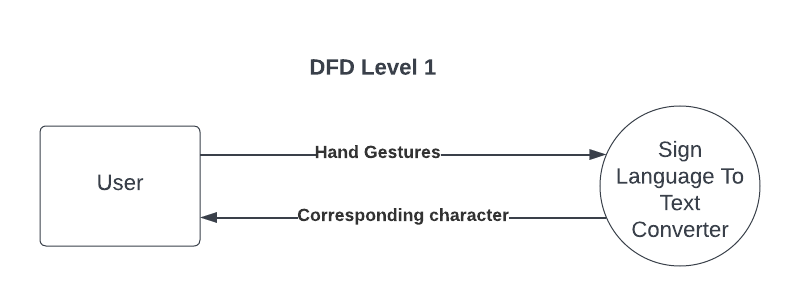
**Fig 3.1 System Flowchart**

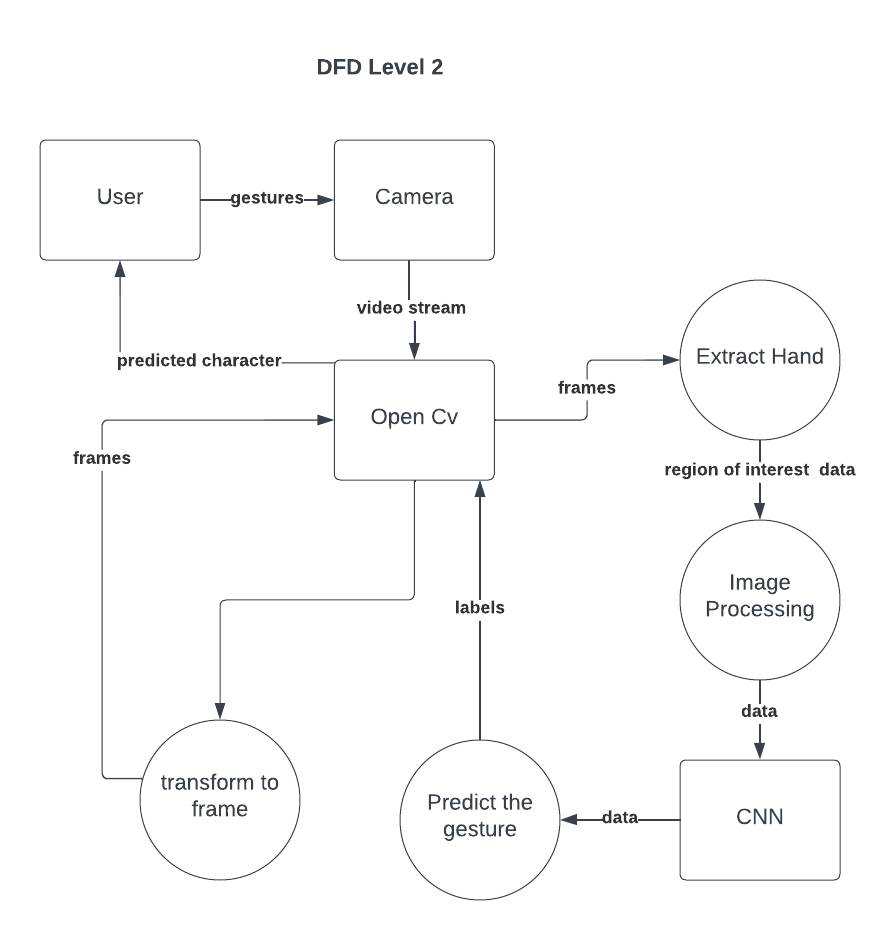
**3.2.2** **Use-case diagram**



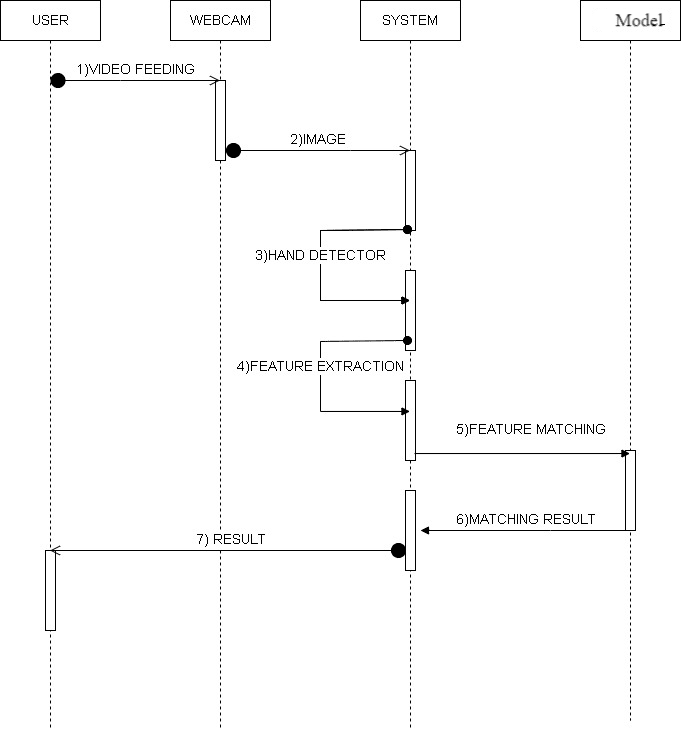
**Fig 3.2 Use Case Diagram**

**3.2.3** **DFD diagram**





**3.2.4** **Sequence diagram**



**Fig 3.3** **Sequence Diagram**

**3.3 CAPTURE**

This is focused on Sign Language (SL), the information of the scene is limited to the hands. To capture hand data a special sensor has been developed to recognize the position of the hand skeleton, Leap Motion controller; compared to other sensors, it is small and portable. It is sometimes attached to a Head Mounted Display to add the user’s hand to a virtual world. Because of physical device constraints, the capture area is limited to a modest distance from the sensor; the area of capture restricts the user movement and it may not be appropriate for sign language.

Hand tracking gloves and markers on the hand are other techniques to capture and track finger information for the recognition of hand gestures. Gloves are accurate in determining the exact position of each finger but they are not comfortable for users, they can obstruct fingers’ movements and the resulting hand gesture may not be correct. Markers are not uncomfortable but they need a specific setup step and a constructed environment. Because casual users of the system may not have in possession the required equipment or they may not know the environment arrangement, this approach may not be suitable for the problem.

**3.4 RECOGNITION**

NNs are popular in image processing over other classification methods because they can, in theory, be trained to perform any regression or discrimination task. These mathematical models are based on reproducing the behaviour of biological brains by learning from real-world data instead of hand-coding features: in the case of image processing, the features of the images.

Nowadays, hardware has more computational power than the early beginning of NNs. Dealing with learning tasks became time-affordable gaining popularity for classification problems. One of the most popular types of NNs for image processing are Convolutional Neural Networks (CNN). They are inspired by the organization of the visual cortex in a biological brain, the perceptrons are not fully connected to the next layer’s perceptrons, instead, they are connected only to some of them. This idea was presented in experiment where it was shown how some specific neurons reacted to particular edge orientations. There are three types of operations in a CNN: the convolution step computes a convolution over the input image using the already learned feature maps as filters; the pooling step reduces the dimensionality of the convolved image; and the fully connected layer works as a common neural network, where each neuron is fully connected to the neurons in the next layer.

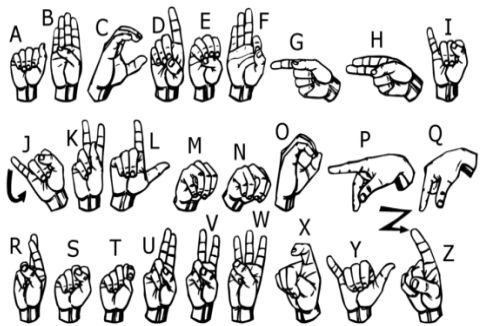
**Chapter 4**

**DATA SETS**

The dataset had to contain images of users sitting in front of the camera performing Sign Language. Images must be captured when the user had performed the symbol; no intermediate gestures were allowed for capture. There must be different users to ensure variations of size and position of gestures in the images.

The main focus of capturing were users’ hands. When the gesture was performed the hand must be visible and trackable. Because we focused on hand letter gesture recognition, hands must be facing towards the camera. Depending on the gesture, fingers may be occluded by the hand. Since in these type of cases occlusions are the characteristics of the gesture they were allowed. If the gesture does not contain finger or palm occlusions in its ideal form, occlusions were not allowed.

The goal of the dataset was to provide gestures in Sign Language for letters of the alphabet from A to Z and symbols for numbers. Sign Language has diverse languages and symbols for letters may vary or may add new ones to the alphabet. The inclusion of the numbers should add more gestures, and thus variety for the database, increasing the scalability of the problem.



**Fig3. The single-handed alphabet signs data set**

Sign Language has 26 letters symbols. The letter are done using finger movement and since we recognized static images, they were not captured. Therefore, the number of letters and numbers remain in 36 symbols. The dataset must be balanced because if it is not, the dataset will be biased towards one specific class.

**4.1 EXTRACTION OF HAND-GESTURE USING DEPTH AS A PARAMETER**

The depth image functionality of the Kinect camera provides us the ability to extract our region of interest, which in our case is the hand gesture, from the rest of the background. However, given the complexity of the ISL hand gestures, the information present in the full frame image is required in order to differentiate the gestures better.

## 4.2 FEATURE EXTRACTION AND HAND-GESTURE RECOGNITION USING DEEP LEARNING

As explained earlier, ISL hand gestures are complex and traditional feature extraction algorithms performs poorly. For example, Canny Edge detection algorithm fails due to the usage of both the hands where edges of one hand can get overlapped or nullified due to the other hand.

Lately, deep learning algorithms has proved to be beneficial in extracting complicated features. Convolutional Neural Network uses the property of convolution, mainly devised for analysing visual imagery. It consists of one input layer and one output layer and numerous hidden layers in between. The hidden layer consists of convolutional layers that compute the dot product between the weights and regions of the input image.

**Chapter 5**

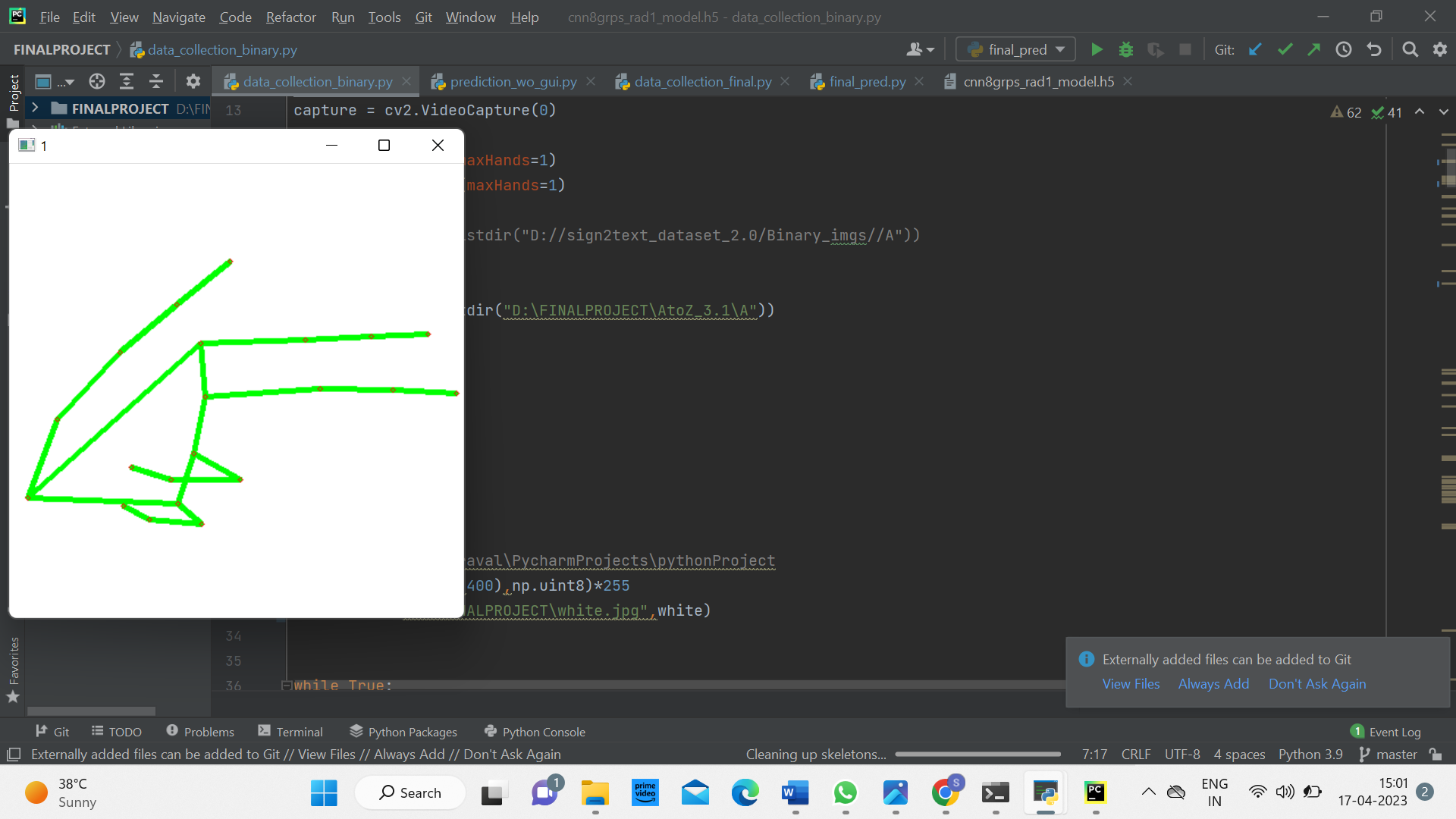
## DESIGN AND IMPLEMENTATION

**5.1 DATA ACQUISITION**

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

It uses electromechanical devices to provide exact hand configuration, and position. Different glove-based approaches can be used to extract information. But it is expensive and not user friendly.

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 costs. 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.

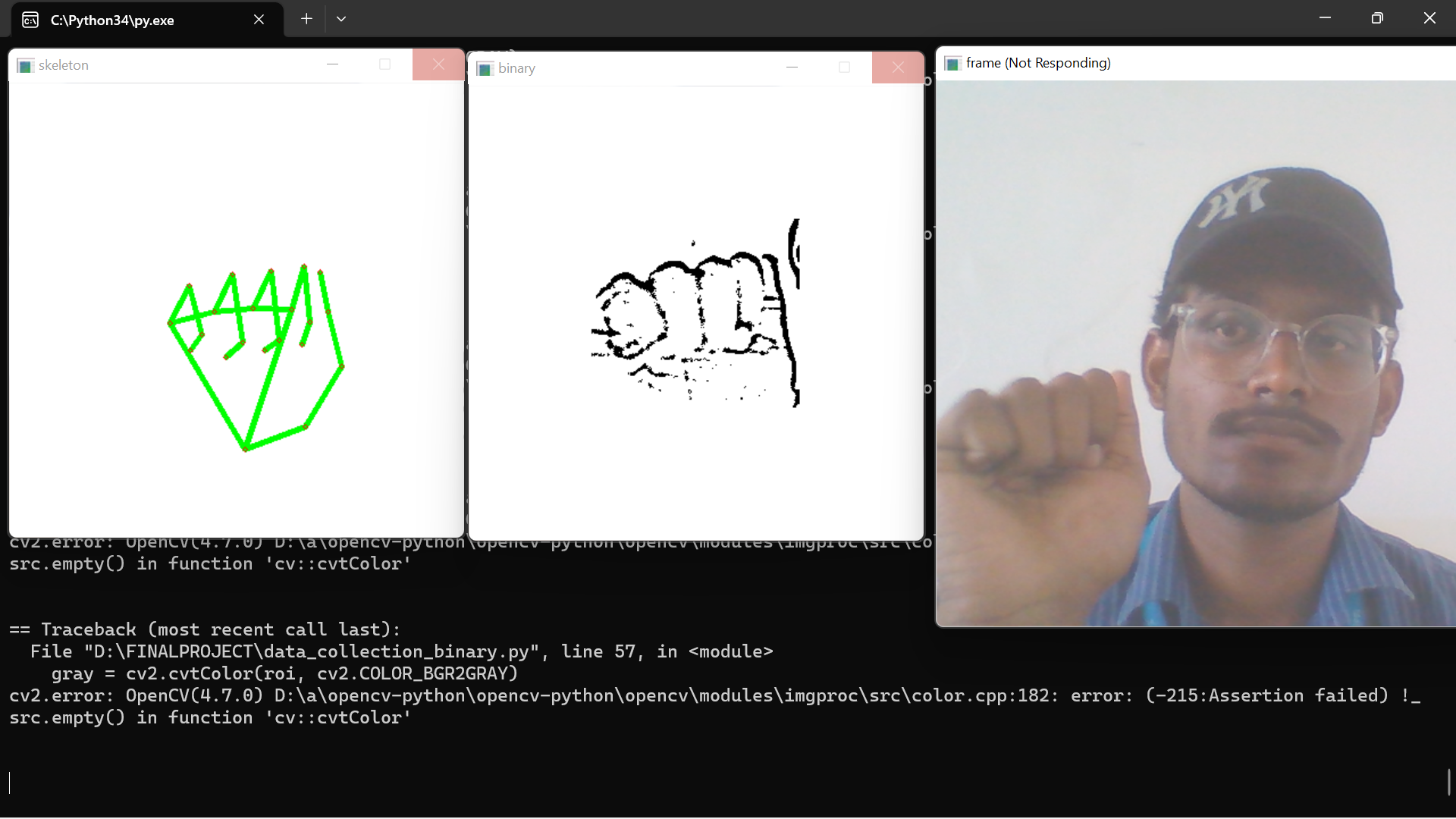


**Fig 5.1 Skeleton Image of Binary Image**

**5.2 DATA PRE-PROCESSING AND FEATURE EXTRACTION**

In this approach for hand detection, firstly we detect hand from image that is acquired by webcam and for detecting a hand we used media pipe library which is used for image processing. So, after finding the hand from image we get the region of interest (Roi) then we cropped that image and convert the image to gray image using OpenCV library after we applied the gaussian blur. The filter can be easily applied using open computer vision library also known as OpenCV. Then we converted the gray image to binary image using threshold and Adaptive threshold methods.

We have collected images of different signs of different angles for sign letter A to Z.

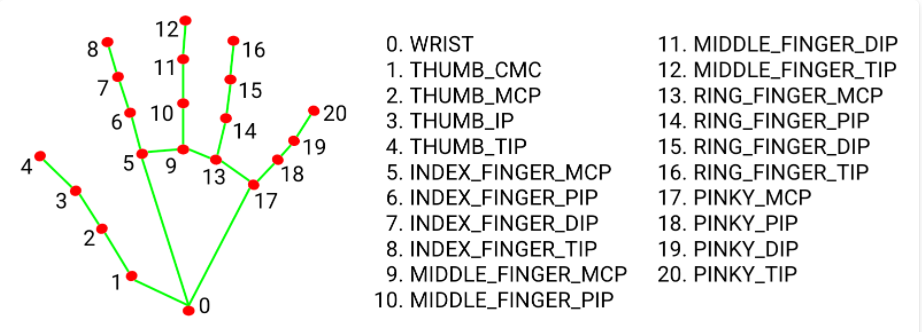


**Fig 5.2 Skeleton and Gray scale image of model**

In this method there are many loop holes like your hand must be ahead of clean soft background and that is in proper lightning condition then only this method will give good accurate results but in real world we don’t get good background everywhere and we don’t get good lightning conditions too.

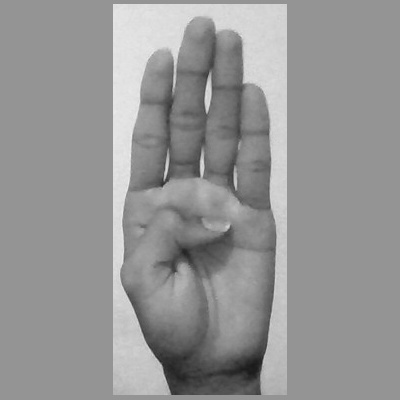
So to overcome this situation we try different approaches then we reached at one interesting solution in which firstly we detect hand from frame using mediapipe and get the hand landmarks of hand present in that image then we draw and connect those landmarks in simple white image.

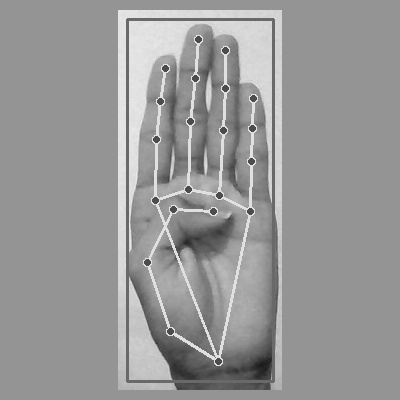
**5.2.1 Landmark System**





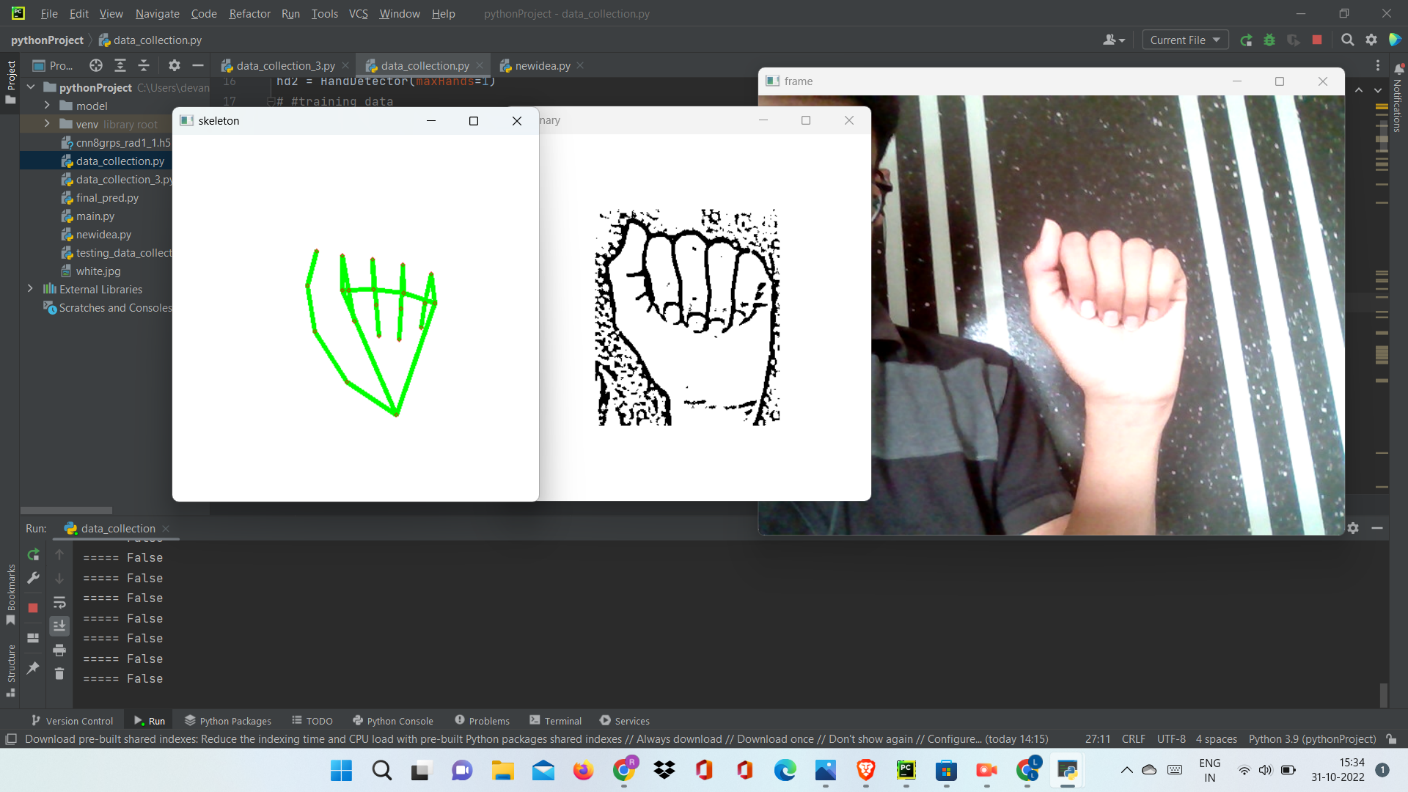
 



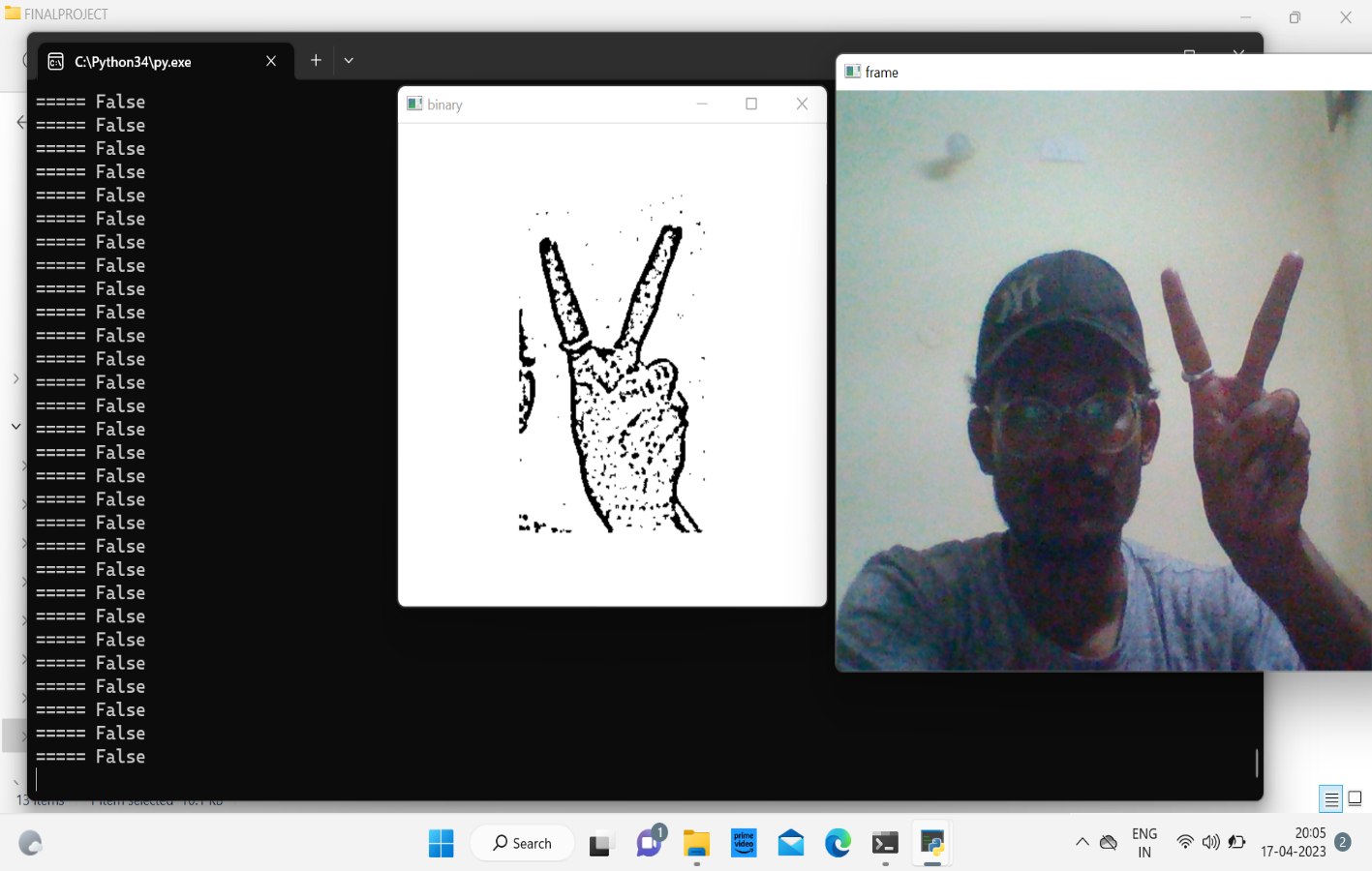
 

Now we get this landmark points and draw it in plain white background using opencv library

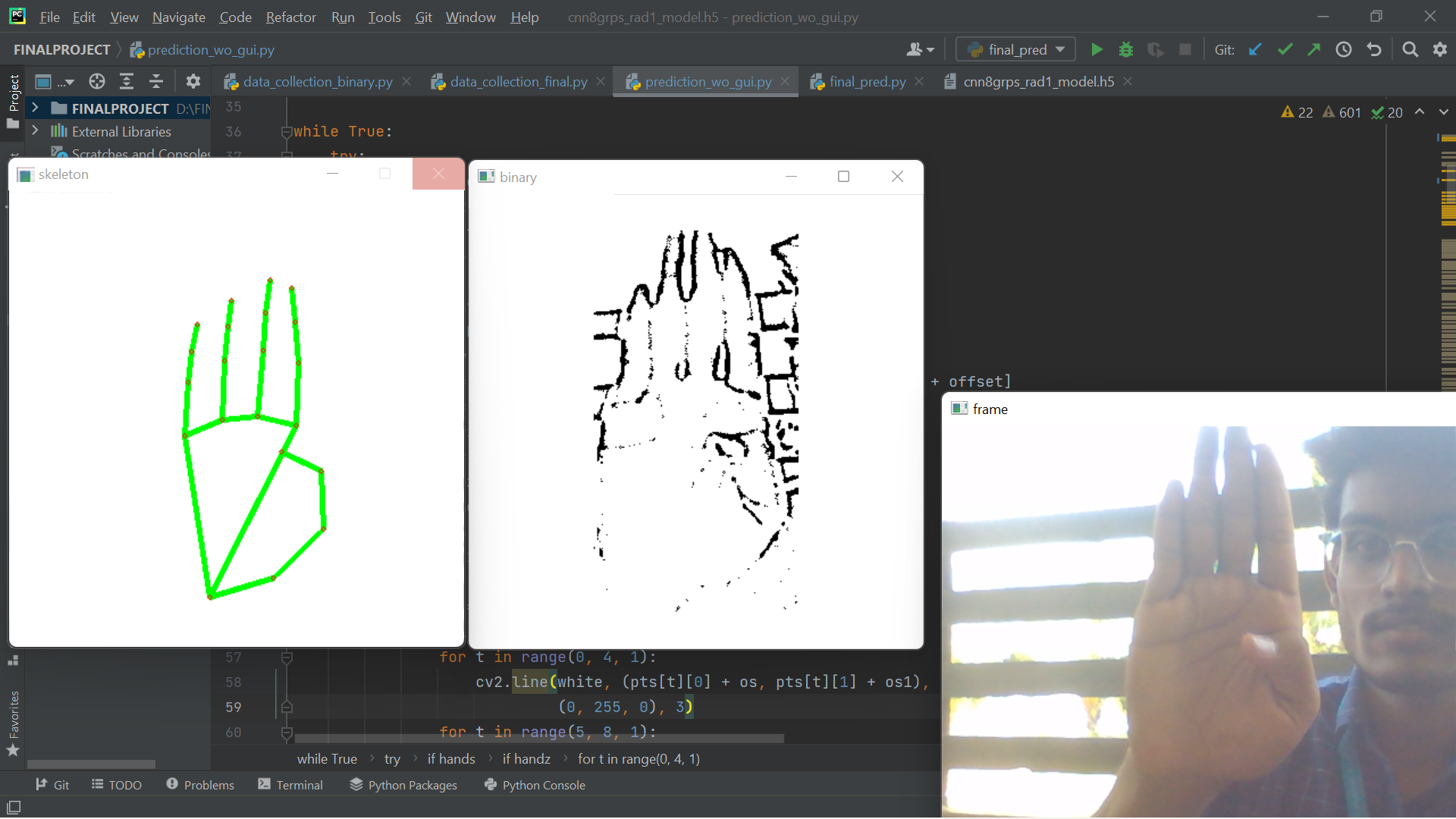
By doing this we tackle the situation of background and lightning conditions because the mediapipe labrary will give us landmark points in any background and mostly in any lightning conditions.



**Fig 5.3 Skeleton And Gray Scale Image Of Alphabet A**



**Fig 5.3 Skeleton And Gray Scale Image Of Alphabet A**



**Fig 5.4** **Skeleton And Gray Scale Image Of Alphabet B**

Collected 720 skeleton images of Alphabets from A to Z

**5.3 GESTURE CLASSIFICATION**

**5.3.1 Convolutional Neural Network (CNN)**

CNN is a class of neural networks that are highly useful in solving computer vision problems. They found inspiration from the actual perception of vision that takes place in the visual cortex of our brain. They make use of a filter/kernel to scan through the entire pixel values of the image and make computations by setting appropriate weights to enable detection of a specific feature. CNN is equipped with layers like convolution layer, max pooling layer, flatten layer, dense layer, dropout layer and a fully connected neural network layer. These layers together make a very powerful tool that can identify features in an image. The starting layers detect low level features that gradually begin to detect more complex higher-level features

Unlike regular Neural Networks, in the layers of CNN, the neurons are arranged in 3 dimensions: width, height, depth.

The neurons in a layer will only be connected to a small region of the layer (window size) before it, instead of all of the neurons in a fully-connected manner.

Moreover, the final output layer would have dimensions (number of classes), because by the end of the CNN architecture we will reduce the full image into a single vector of class scores.

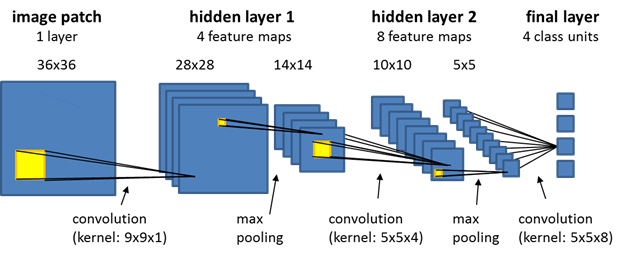
**Convolutional Layer:**

In convolution layer I have taken a small window size [typically of length 5\*5] that extends to the depth of the input matrix.

The layer consists of learnable filters of window size. During every iteration I slid the window by stride size [typically 1], and compute the dot product of filter entries and input values at a given position.

As I continue this process well create a 2-Dimensional activation matrix that gives the response of that matrix at every spatial position.

That is, the network will learn filters that activate when they see some type of visual feature such as an edge of some orientation or a blotch of some colour.



**Pooling Layer:**

We use pooling layer to decrease the size of activation matrix and ultimately reduce the learnable parameters.

There are two types of pooling:

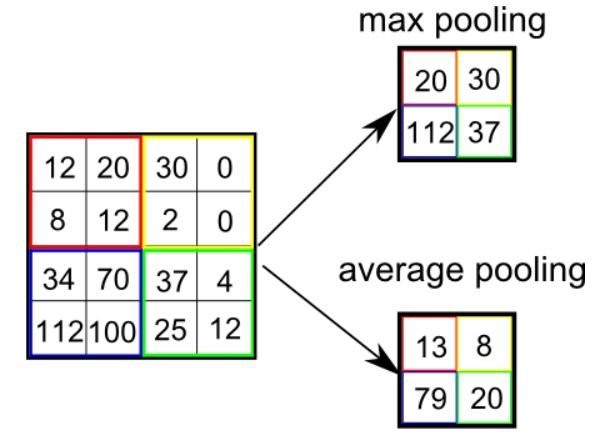
**a. Max Pooling:**

In max pooling we take a window size [for example window of size 2\*2], and only taken the maximum of 4 values.

Well lid this window and continue this process, so well finally get an activation matrix half of its original Size.

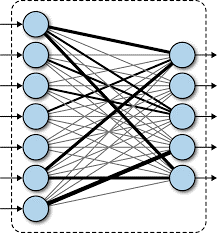
**b. Average Pooling:**

In average pooling we take average of all Values in a window.



In convolution layer neurons are connected only to a local region, while in a fully connected region, well connect the all the inputs to neurons.

**Fully Connected Layer**



The preprocessed 180 images/alphabet will feed the keras CNN model.

Because we got bad accuracy in 26 different classes thus, We divided whole 26 different alphabets into 8 classes in which every class contains similar alphabets: [y,j]

[c,o]

[g,h]

[b,d,f,I,u,v,k,r,w]

[p,q,z]

[a,e,m,n,s,t]

All the gesture labels will be assigned with aprobability. The label with the highest probability will treated to be the predicted label.

So when model will classify [aemnst] in one single class using mathematical operation on hand landmarks we will classify further into single alphabet a or e or m or n or s or t.

**Text To Speech Translation**

The model translates known gestures into words. we have used pyttsx3 library to convert the recognized words into the appropriate speech. The text-to-speech output is a simple workaround, but it's a useful feature because it simulates a real-life dialogue.

**5.3.2** **TENSORFLOW**

TensorFlow is an end-to-end open-source platform for Machine Learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in Machine Learning and developers easily build and deploy Machine Learning powered applications.

TensorFlow offers multiple levels of abstraction so you can choose the right one for your needs. Build and train models by using the high-level Keras API, which makes getting started with TensorFlow and machine learning easy.

If you need more flexibility, eager execution allows for immediate iteration and intuitive debugging. For large ML training tasks, use the Distribution Strategy API for distributed training on different hardware configurations without changing the model definition.

**5.3.3 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.

**5.3.4 OPENCV**

OpenCV (Open-Source Computer Vision) is an open-source library of programming functions used for real-time computer-vision.

It is mainly used for image processing, video capture and analysis for features like face and object recognition. It is written in C++ which is its primary interface, however bindings are available for Python, Java, MATLAB/OCTAVE.

**Chapter 6**

## RESULTS AND DISCUSSION

When layer 1 and layer 2 are combined, project accuracy increases to 98.0%, surpassing the accuracy of the majority of recent research articles on sign language. This project obtained an accuracy of 95.8% in model using only layer 1 of method.

The majority of research articles concentrate on employing tools like Kinect to detect hands.

Convolutional neural networks (CNN) and Kinect are used to construct a sign language recognition system, which achieves an error rate of 2.5%.

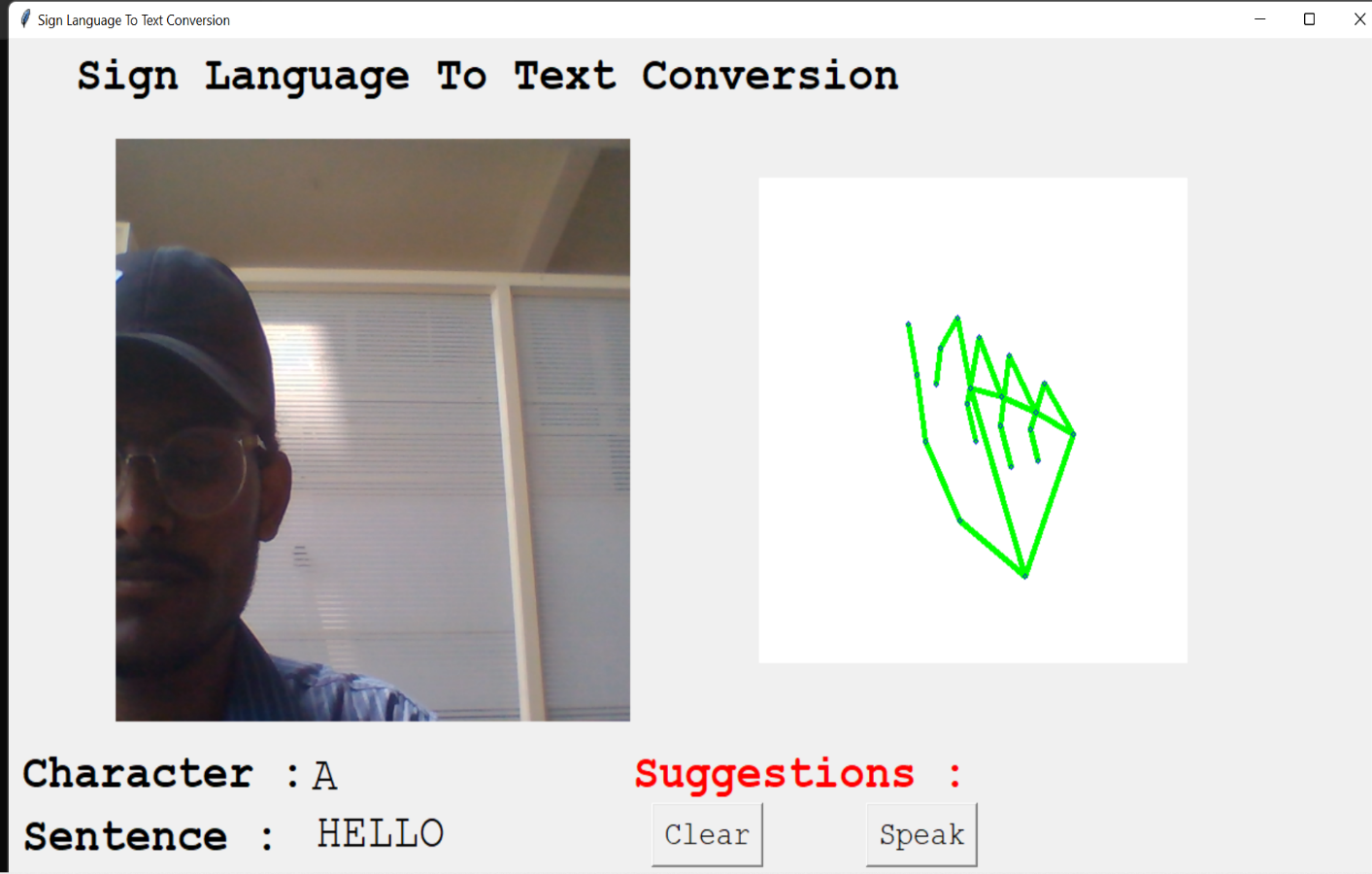
Bottom of Form

Sign language recognition has been a challenging research area due to the complexity and variability of sign language gestures. While some research has focused on using devices like Kinect for hand detection, others have used computer vision techniques such as convolutional neural networks (CNNs) to recognize signs.

In recent years, deep learning techniques like CNNs have shown promising results in sign language recognition. However, the performance of these models heavily depends on the availability of large annotated datasets, which can be difficult to obtain for sign languages with fewer speakers.

One approach to address this challenge is to leverage transfer learning, where a pre-trained CNN model is fine-tuned on a smaller sign language dataset. Another approach is to use generative adversarial networks (GANs) to generate synthetic sign language data, which can be used to augment the training dataset.

Overall, sign language recognition has the potential to enable greater communication and accessibility for the deaf and hard-of-hearing communities. Continued research in this area can help improve the accuracy and robustness of sign language recognition systems, making them more accessible and useful for people around the world.



They also used CNN for their recognition system. One thing should be noted that our model doesn’t uses any background subtraction algorithm whiles some of the models present above do that.

So, once we try to implement background subtraction in our project the accuracies may vary. On the other hand, most of the above projects use Kinect devices but our main aim was to create a project which can be used with readily available resources. A sensor like Kinect not only isn’t readily available but also is expensive for most of audience to buy and our model uses a normal webcam of the laptop hence it is great plus point.

**Chapter 7**

## CONCLUSION AND FUTURE SCOPE

**7.1 CONCLUSIONS**

This project included a real-time, automatic sign language gesture recognition technology. Recognising Sign Language (SL) from photographs is still a difficult subject. Ambiguity is caused by similar movements, user accents, context, and signs with different meanings. These are a few reasons why earlier research only used small datasets. We came to the conclusion that a dataset for training and testing the system must comprise enough gesture variants for each signal to be generalised. We improved the average approach to assess whether a dataset contains a sufficient variety of motions, and we constructed a Convolutional Neural Network (CNN) for identifying hand gestures in pictures of ASL letter and number symbols.

There was just one external dataset that met both of my criteria for doing experiments: Images must have depth, and the hand must be clearly seen.

The CNN method outperformed the typical approach in terms of recognition rate. Because some of the letters in Sign Language have extremely similar symbols and the dataset has a large variety of gesture shapes, the algorithm had difficulty correctly classifying some of the photos. CNN mislabeled several of the photographs because they resembled one another. If the right classification was among the top three rather than the top one, it signified that the system came close to correctly classifying the image, but CNN misclassified it as a result of similar symbols.

## 7.2 FUTURE SCOPE

1. A model for sign language word and sentence level recognition can be created. A system that can recognise changes in the temporal space will be needed for this.

2. By creating a comprehensive solution, we can bridge the communication gap for those who are deaf or hard of hearing.

**BIBLIOGRAPHY**

[1] Brill R. 1986. The Conference of Educational Administrators Serving the Deaf: A History. Washington, DC: Gallaudet University Press.

[2] P. K. Athira, C. J. Sruthi, A. Lijiya, “A Signer Independent Sign Language Recognition with Co-articulation Elimination from Live Videos: An InScenario”,2019.https://www.sciencedirect.com/science/article/pii/S131915781831228X

[3] M. Jaiswal, V. Sharma, A. Sharma, S. Saini, R. Tomar, “An Efficient Binarized Neural Network for Recognizing Two Hands Indian Sign Language Gestures in Real-time Environment”,2020https://ieeexplore.ieee.org/abstract/document/9342454

[4] C. J. Sruthi, A. Lijiya, “Signet: A Deep Learning based Indian Sign Language Recognition System”,2019.https://ieeexplore.ieee.org/abstract/document/8698006

[5] T. Raghuveera, R. Deepthi, R. Mangalashri, R. Akshaya, “A depth-based Indian Sign Language recognition using Microsoft Kinet <https://link.springer.com/article/10.1007/s>12046-019-1250-6.

[6].A. Wadhawan, P. Kumar, “Deep learning-based sign language recognition system for static signs”, 2020.https://link.springer.com/article/10.1007/s00521-019-04691-y

[7] S. Z. Gurbuz, A. C. Gurbuz, E. A. Malaia, D. J. Griffin, C. S. Crawford, M. M. Rahman, E. Kurtoglu, R. Aksu,T. Macks, R. Mdrafi, “American Sign Language Recognition Using RF Sensing”, 2020.https://ieeexplore.ieee.org/abstract/document/9187644

[8] M. Al-Hammadi, G. Muhammad, w. Abdul, M. Alsulaiman, M. A. Bencherif, M. A. Mekhtiche, “Hand Gesture Recognition for Sign Language Using 3DCNN”, 2020.https://ieeexplore.ieee.org/abstract/document/9078786

[9] R. Rastgoo, K. Kiani, S. Escalera, “Video-based isolated hand sign language recognition using a deep cascaded model”, 2020.https://link.springer.com/article/10.1007%2Fs11042-020-09048-5

[10] G. A. Rao, K. Syamala, P. V. V. Kishore, A. S. C. S. Sastry, “Deep Convolutional Neural Networks for Sign Language Recognition”, 2018. https://ieeexplore.ieee.org/abstract/document/8316344.