Bachelor's Term Project-I (EC47003) report on

"HAND SIGN RECOGNITION"

submitted in partial fulfillment of the requirements

for the award of the degree of

B.Tech. (Interdisciplinary Dual Degree Program 5Y)

In

Electronics and Electrical Communication Engineering

by

SAI SUNDARA DINESH YELLETI

(20EC39031)

Under the guidance of

Prof. VIVEK DIXIT

DEPARTMENT OF ELECTRONICS AND ELECTRICAL COMMUNICATION ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

2022-2023

DECLARATION

I certify that

(a) The work contained in this report has been done by me under the guidance of my supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever I have used materials (data, theoretical analysis, figures, and text)from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Date: ______________
Place: Kharagpur

(Sai Sundara Dinesh Yelleti)
(20EC39031)

DEPARTMENT OF ELECTRONICS AND ELECTRICAL COMMUNICATION INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR - 721302, INDIA

CERTIFICATE

This is to certify that the project report entitled "**Hand Sign Recognition**" submitted by **Sai Sundara Dinesh Yelleti** (Roll No. 20EC39031) to Indian Institute of Technology Kharagpur towards partial fulfillment of requirements for the award of degree of Bachelor of Technology in Electronics and Electrical Communication Engineering is a record of bonafide work carried out by him under my supervision and guidance during Autumn Semester, 2022-23.

Date: _____________

Place: Kharagpur

Dr. Vivek Dixit
Professor

Department of Electronics and Electrical Communication
Indian Institute of Technology, Kharagpur

ABSTRACT

Sign language is a visual and gestural language used by deaf and hard-of-hearing individuals to communicate. Instead of spoken words, sign languages rely on hand movements, facial expressions, body language, and gestures to convey meaning. Different countries and regions may have their own sign languages, each with its own grammar and vocabulary. Sign language is a complete and natural language, recognized as such by linguistic studies, and it plays a crucial role in facilitating communication and fostering community among individuals with hearing impairments. This paper offers a system for recognizing hand gestures by processing images and predicting motions using a deep learning algorithm, Convolution Neural Network (CNN). The research on hand gesture recognition in sign language has profound implications for the fields of human-computer interaction (HCI) and assistive technology. Integrating hand gesture recognition in sign language into HCI and assistive technology significantly enhances accessibility for individuals with hearing impairments. This article demonstrates the recognition of 26 alphabets and 0-9 digit hand movements in American Sign Language. The proposed system includes pre-processing and feature extraction modules, model training and testing, and sign-to-text conversion. Different CNN architecture and pre-processing techniques, such as Tensors, Landmarking, and Dynamic Aspect ratio, were designed and tested with our dataset to obtain better accuracy in recognition. The proposed system for recognizing hand gestures in sign language has the potential to extend its applications beyond the recognition of alphabets and digits, opening up new possibilities in communication and accessibility.

Keywords: Sign Language, Deep learning, Convolution Neural Networks(CNN), Human-Computer Interaction, American Sign Language (ASL)

CONTENTS

CHAPTER-1

Introduction

Individuals who experience complete or partial damage to their hearing aids are categorized as deaf. To communicate with those who possess functional hearing aids, deaf individuals often resort to lip-reading or sign language—a form of communication utilizing hand signs, facial expressions, and body postures predominantly employed by the deaf community. Sign languages vary across regions, with examples including Indian and American Sign. Additionally, individuals may invent hand movements for communication when faced with language barriers.

Unfortunately, those with normal hearing often lack awareness of sign language, resulting in communication gaps with deaf individuals. Communication, defined as the exchange of ideas or messages through gestures or text, underscores the importance of hand gestures as a potent means of conveying thoughts. Gestures can be either static, involving hand shape, or dynamic, encompassing hand movements during communication.

Sign language emerges as the fundamental and innate mode of communication for the deaf and hard of hearing. Unfortunately, society tends to neglect and isolate physically impaired individuals. Bridging the communication gap between deaf people and the general population necessitates familiarity with sign language. The objective is to establish an efficient system overcoming this communication barrier, enabling muted individuals to communicate with those who do not understand sign language without the aid of an interpreter.

This project's focal point is the recognition of American Sign Language through software using Python Language. The aim is to give an image-based, low-cost, real-time application device that could easily recognize sign language and hand gestures. Limitations of the image-based systems are strictly related to the quality of the camera used and its adaptability to different lighting conditions. The system, centered around 26 hand gestures of American Sign Language, utilized a dataset comprising around 400 images per alphabet and 400 images per number. The primary task involved preprocessing the images, incorporating algorithms such as greyscaling(optional), landmarking, cropping, and dynamically adjusting the aspect ratio for training.

Following preprocessing, the images underwent training and testing within a Convolutional Neural Network (CNN) model. Various CNN architectures were implemented and tested on the dataset to determine the most influential architecture for recognizing hand gestures.

1.1 Motivation

Empowerment of the deaf community is a driving force behind sign language recognition projects. Effective communication is a cornerstone of individual empowerment, and these projects seek to provide tools that enable deaf individuals to express themselves fully. The recognition systems contribute to active participation in various facets of life, promoting autonomy and empowerment. Recognizing the crucial role sign language plays in education for deaf people, this project is also motivated by a desire to provide educational support. Creating tools that facilitate the learning and understanding of sign language enhances educational experiences for deaf and non-deaf individuals. This educational support contributes to a more inclusive and enriching learning environment. Also, this project serves as a vehicles for raising awareness and understanding about the challenges faced by the deaf community. By developing and implementing these systems, there is a concerted effort to encourage a broader societal understanding of the importance of inclusive communication. This awareness fosters a more compassionate and empathetic society.

1.2 Contribution

The main contributions of this work can be stated as follows:

1. Our exploration encompassed a detailed understanding of the different types of sign languages used worldwide and how letters and numbers are represented in ASL.
2. A sophisticated code was developed, seamlessly integrating Mediapipe and OpenCV to extract precise hand gestures from the input video, showcasing the technical prowess of our approach.
3. Implementation of this code facilitated the collection of data using Mediapipe for landmarking the hand and TensorFlow for model detection.
4. Rigorous training was conducted, underscoring the superior accuracy and performance of the proposed code, affirming its efficiency in the context of hand sign detection.

CHAPTER-2

Sign Language Symbols

In American Sign Language (ASL), signs are categorized based on the number of hands involved in their formation. The system distinguishes between one-handed signs, where communication is facilitated using a singular hand, and two-handed signs, which necessitate the coordinated movement of both hands. This classification system extends to the intricate combination of handshapes, movements, locations, and non-manual markers, such as facial expressions and body postures, all of which collectively contribute to the nuanced and expressive grammar of ASL.

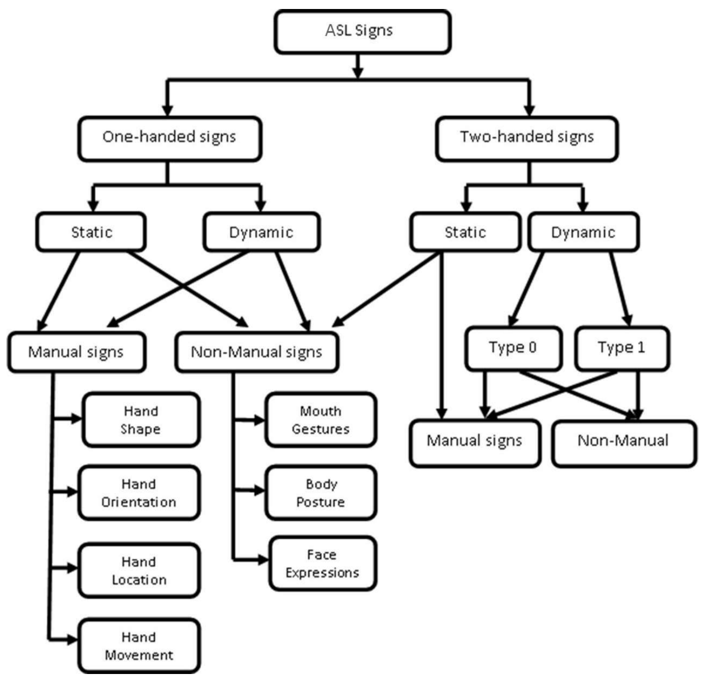


Figure 1. Hierarchy of Signs

2.1 One-Handed Signs

One-handed signs in American Sign Language (ASL) encompass a diverse range of gestures and expressions executed using a single hand. For representing one-handed signs, a single dominant hand is used. These signs involve specific handshapes, movements, locations, and orientations to convey distinct meanings. Examples of one-handed signs in ASL include the formation of letters, numbers, and various lexical items through the manipulation of a solitary hand in different ways. The richness of ASL lies in the intricate combinations of these elements, allowing for the conveyance of nuanced and context-dependent messages with a single hand.

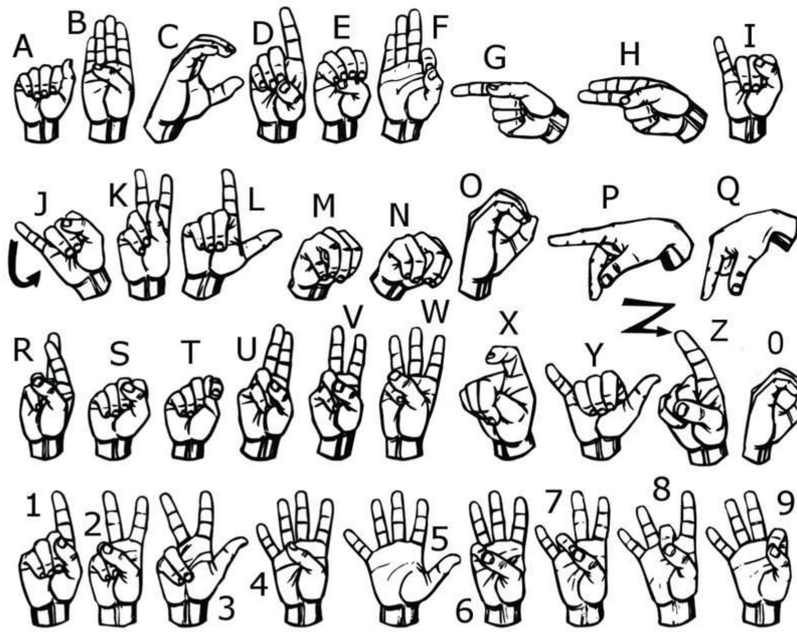
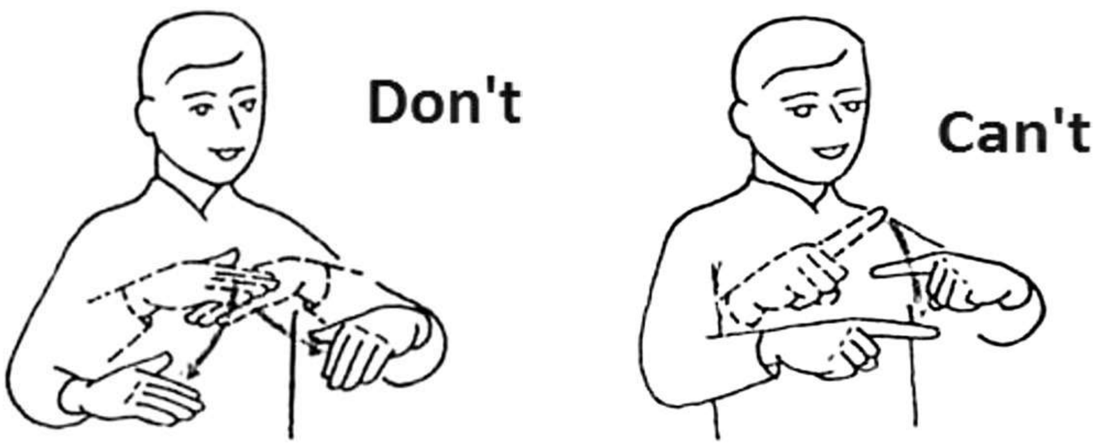


Figure 2. American Sign Language for alphabets and numbers

2.1 Two-Handed Signs

Two-handed signs in American Sign Language (ASL) involve the coordinated movement and positioning of both hands to convey meaning. This category encompasses a wide array of signs that utilize simultaneous gestures with both hands to express concepts, words, or phrases. Two-handed signs may involve symmetrical or asymmetrical handshapes, dynamic movements, and specific spatial relationships between the hands. This dual-handed approach in ASL contributes to the language's capacity for expressing complex and nuanced ideas through the simultaneous use of both hands, further enhancing the richness and versatility of communication in sign language.



(a) **(b)**

Figure 3. a) Two-handed type 0 sign (both hands are active), b) Two-handed type 1 sign (only dominant hand is active)

CHAPTER-3

Related Work

This section summarizes the existing work on hand sign detection in this research domain. The issue of detecting hand gestures in sign languages has been a subject of extensive global research, with various proposed approaches distinguished primarily by the Datasets and Features looked at in the datasets.

Sign Language recognition can be categorized into different sections. Most of the time, the categorization of sign language hand gestures is captured by the system. First, if we consider which type of system we both want to use, we can decide how to proceed with implementing the system. The vision-based system (where the gestures are captured using one or more cameras), and the device-based system (where a direct-measure device such as designated electronic gloves equipped with sensors are generally employed to connect the user with the system).[4]

While the device-based systems are characterized by their efficiency, their real-life usage is limited due to the need to wear the cumbersome device when interacting with the system. This issue, however, does not arise for vision-based systems, allowing users to interact more naturally with the system [4][7]. In terms of applicability, it has a wider application in outdoor scenarios.

This ease in using the vision-based system was challenged by how it handles datasets made up of dynamic hand gestures in sign language, such as isolated and continuous signs. According to [5], while most existing works focus on recognizing isolated gestures, their use in real-world applications is limited. Moreover, developing hand gesture recognition using the vision-based system requires more powerful feature extraction and discrimination methods [4][8].

The interest in gesture recognition has led to a large body of research, as has been noted in several review papers [4][13], [14]–[17]. Cheok et al. [4][13] reviewed the state-of-the-art technique used in recent hand gesture and sign language recognition research in data acquisition, pre-processing, segmentation, feature extraction, and classification. Wadhawan et al. [4][3] focused on academic literature published from 2007–2017. These papers were reviewed in six dimensions (data acquisition techniques, static/dynamic signs, signing mode, single/double-handed signs, classification techniques, and recognition rates). Recently, Aloysius and Geetha [13] reviewed the vision-based continuous sign language recognition (CSLR) system. Ratsgoo et al. [14] focused on the vision-based proposed models of sign language recognition

Priyanka C Pankajakshan and Thilagavati B [1][6] have proposed a system for sign language recognition using ANN (Artificial Neural Network). The system consists of live capture of hand gestures to process and identify the sign using ANN. Das, A., Gawde, S., Suratwala, K., & Kalbande .D. [1][8] and Rao, G. A., Syamala, K., Kishore, P. V. V., & Sastry. [1][9] has performed fundamental research on sign language datasets using the CNN algorithm to achieve satisfactory results from the training and testing of the dataset. In [1][8], the authors proposed a system using CNN's Inception v3 on the dataset to test its accuracy and found it to be better than CNN. In [1][9], the proposed system uses selfie language to process the images and is tested using stochastic pooling. The CNN model used to train the dataset in [1][9] was performed using different window sizes and different batches of images, and the output accuracy achieved by them was 92.88% compared to the other methods they researched.

Anup Kumar, Karun Thankachan, and Mevin M. Dominic [1][18] have developed a system using Support Vector Machine. The images used in [1][18] are pre-processed using skin segmentation, and appropriate features from the image obtained are extracted. This pre-processing of images is completed by converting the image to greyscale and then performing HSV thresholding [1][18].

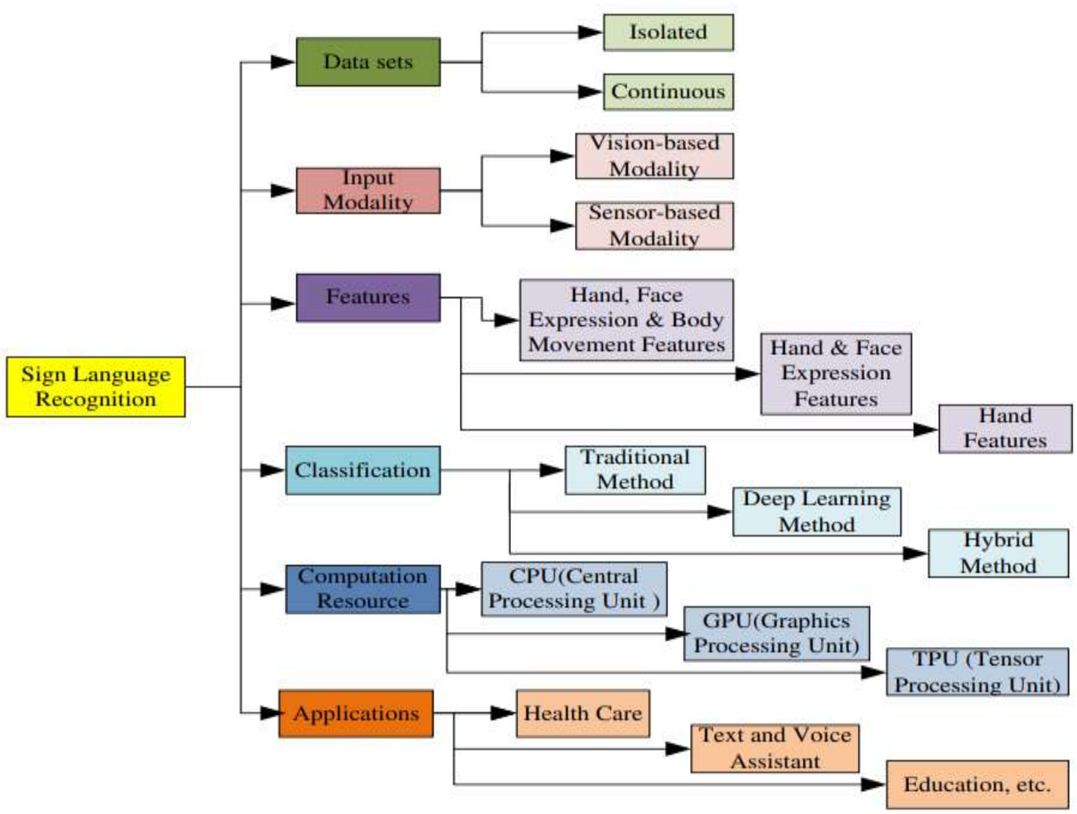


Figure 4. : The Fundamental Attributes of SLR[2]

CHAPTER-4

Methodology

The module below is fed an image as an input that delivers the alphabet/number corresponding to the sign visible in the image/video. The process involves four major steps:

1. Pre-processing the image
2. Data Collection
3. CNN Model training
4. Feeding the new image into the CNN model
5. Displaying the predicted text for the image passed

In the first step; the input is obtained as an image or captured from a video. The captured image is then preprocessed using the methods mentioned in Chapter 4 and forwarded to the CNN model. The CNN model tests the loaded image against the trained images and predicts the sign with the most probable labels from the already trained model.

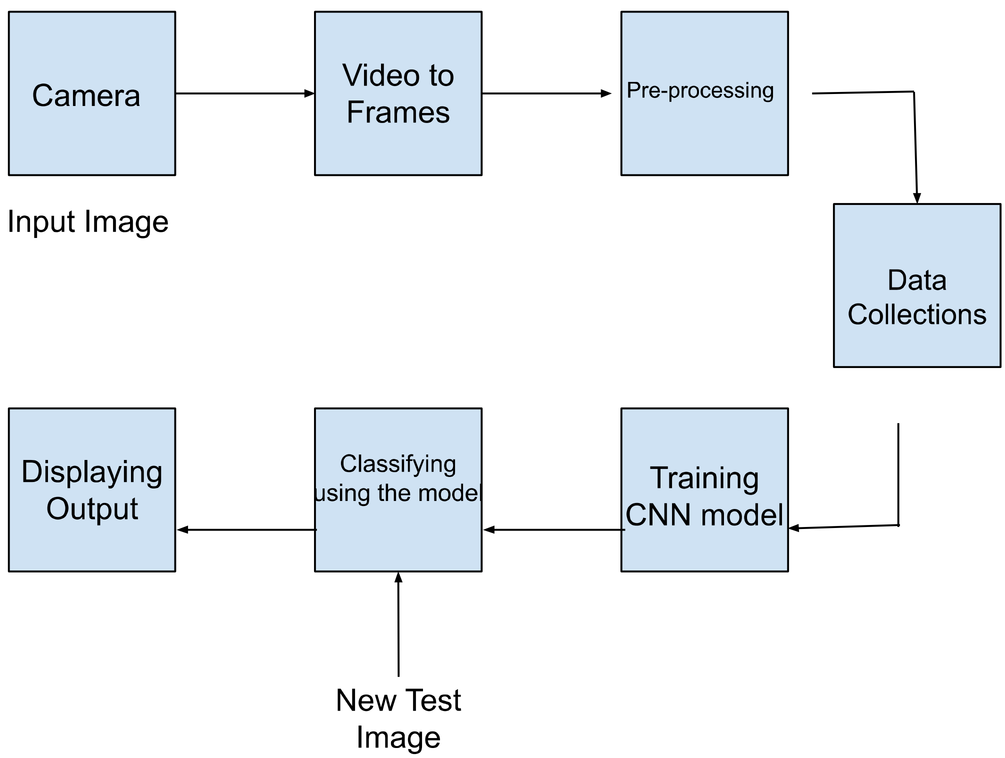


Figure 5. Pipeline for Hand Gesture Recognition

4.1 Pre-processing

CNN model cannot be trained and cannot accurately predict labels when unprocessed images are fed directly to it. The main issue with CNN processing is its inability to cancel out the background properly. Hence, the images must be processed separately using various image processing techniques.

At first, the complete image from the webcam looks like this -

Figure 6. Complete image we get on webcam

Here, instead of the whole image, we just need the Hand part for the training of the model -

Figure 7. Required image for training

If the whole image is given to train the model, the CNN will also take the background features, which, in this case, is unnecessary. The input layer should not contain unnecessary data for the CNN model to work correctly. Including irrelevant information in the background may introduce noise and distractions, making it harder for the model to focus on the relevant features for hand gesture recognition. Therefore, we must detect the hand and crop the hand out from the whole image. For this purpose, we use Mediapipe.

4.1.1 Mediapipe

Mediapipe, an open-source library developed by Google, is a versatile framework for real-time computer vision applications. This powerful tool simplifies the implementation of various perception tasks, offering pre-built modules for tasks like hand tracking, face detection, and pose estimation. Its modularity allows developers to use specific components independently and boasts cross-platform compatibility, making it adaptable to diverse devices. With a flexible pipeline and efficient, pre-trained models, Mediapipe empowers developers to create applications with real-time visual understanding, particularly excelling in hand gesture recognition and face expression understanding tasks. Its user-friendly interface and robust capabilities make it a valuable asset for projects requiring rapid computer vision prototyping and implementation.

For this project, we will be using a hand-detection module. The module was designed to accurately detect and track hands in real-time from video feeds or image sequences. This module employs a machine learning model trained on hand images to identify key landmarks and track the movement of the hand.

Figure 8. Detection of hand using Mediapipe

After getting the landmarks visible on the hand, we can get the positions of the landmarks from the image. By acquiring the maximum and minimum values of x coordinates and y coordinates from the landmark positions, we get the boundary box of the hand. The term "boundary box" refers to a bounding box, a rectangular box that encapsulates an object or region of interest within an image. We get the exact bounding box containing our whole palm by setting an optimum offset.

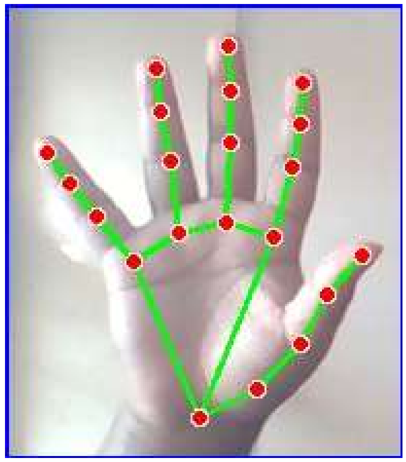


Figure 9. Cropped Image of hand with landmarks.

Now, there comes a problem. Few signs are made spreading the fingers wide across, and few are made putting them straight up. Due to this, there will be variations in the sizes of cropped images. Training a model with irregular data can give incorrect results. Therefore, we have to adjust the image to equal sizes.

4.1.2 Aspect Ratio

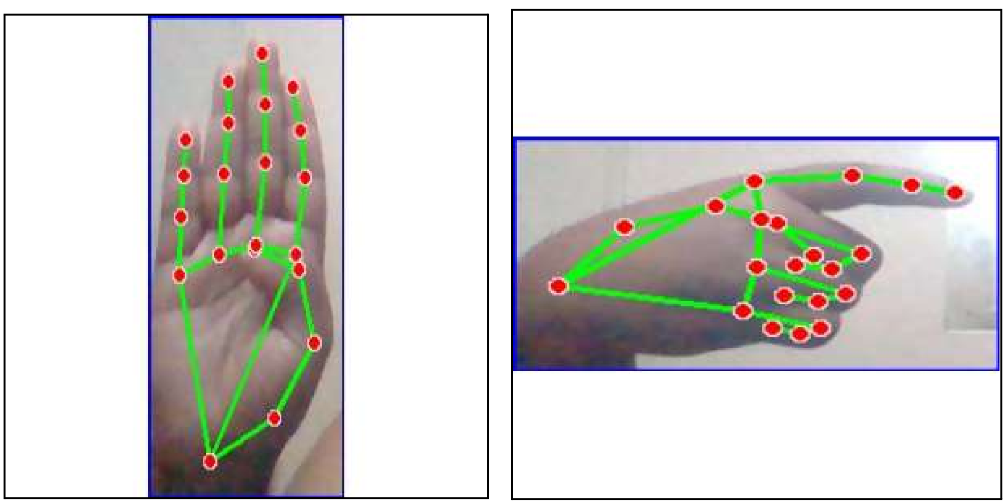
Aspect ratio is the proportional relationship between the width and height of an image. The aspect ratio of an image is a crucial property because it defines the proportional relationship between its width and height. Changing the aspect ratio can significantly impact how the image is perceived and displayed. Therefore, we should only try to change the aspect ratio of an image with necessity. We have the image in irregular shapes; few are long, and few are broad. We must change the image shape and resize it without changing its aspect ratio.
We change the image size by placing them on a blank white screen of size 300x300 using few lines of code:

```
aspect_ratio = h/w
                if aspect_ratio>1:
                    k = 300/h
                    wcal = math.ceil(k * w)
                    wgap = math.ceil((300 - wcal)/2)
                    cropImage_resize =
```

```
cv2.resize(hand_img_crop,(wcal, 300))
                    white_screen[0:300,wgap:wgap+wcal] =
cropImage_resize

                else:
                    hcal = math.ceil(300*aspect_ratio)
                    hgap = math.ceil((300 - hcal)/2)
                    cropImage_resize = cv2.resize(hand_img_crop,(300,
hcal))
                    white_screen[hgap:hgap+hcal,0:300] =
cropImage_resize
```

By applying this code, we achieve a dynamic image reshaping without changing the aspect ration of the image.



(a) **(b)**

Figure 10. Images of sign a)B and b)G after adjusting them to equal size

4.1.3 Grey Scaling

The last method we used, which is optional, to process our images was by just converting the RGB image to Greyscale and resizing the image to pass through the CNN model used to train the images. Grayscale is a range of consistent monochrome shades from black to white. Digital images can be saved as black and white images, even color images comprised of grayscale information. Each pixel contains a luminance value, regardless of its color. Luminance for images can be characterized as brightness or intensity, which can be measured on a scale from black to white.

$$\text{Pixel} = 0.299 * \text{Red} + 0.587 * \text{Green} + 0.114 * \text{Blue}$$

4.1 Model Training

Training a model for images involves using machine learning algorithms to teach a computer how to recognize patterns and features within images. There are two types of machine learning algorithms: Supervised and Unsupervised learning algorithms. In unsupervised learning, the algorithm is given data without explicit instructions on what to do with it. The data is unlabeled, and the algorithm must find its own patterns, structures, or relationships within the data. The goal is often to discover the data's inherent structure, such as clusters or associations, without predefined labels. In supervised learning, the algorithm is trained on a labeled dataset. Each training set example includes input data and the corresponding correct output (label). The goal is to learn a mapping from inputs to outputs so that the algorithm can make accurate predictions or classifications on new, unseen data.

Neural networks are computational models inspired by the structure and functioning of the human brain, designed to perform complex tasks by learning patterns from data. Composed of interconnected nodes, or artificial neurons, organized into hidden layers, neural networks excel at capturing intricate relationships in diverse datasets. Input data is fed into the network, and through a process of weighted connections and activation functions, the network transforms this input into meaningful output. We train our model using Convolutional Neural Networks, a Deep learning algorithm.

4.2.1 Convolutional Neural Network(CNN)

We have designed our CNN algorithm based on the knowledge perceived from [8][9]. CNNs are a fundamental example of deep learning, where a more sophisticated model pushes the evolution of artificial intelligence by offering systems that simulate different biological human brain activity types and are fed with processed images to classify the images.

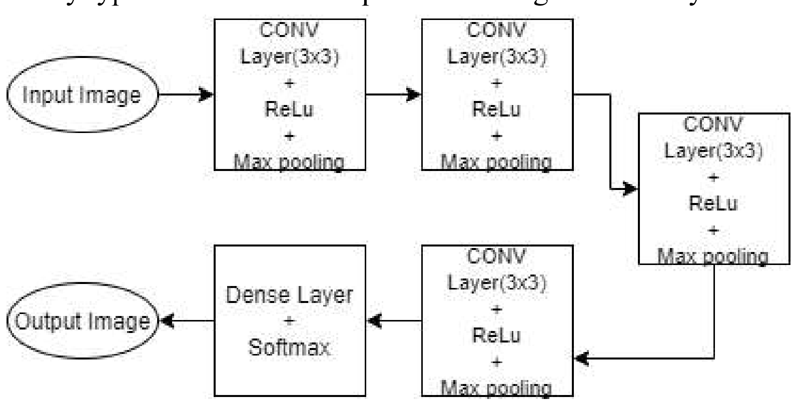


Figure 11. Proposed Deep CNN Architecture

Various interpretations were used to measure the performance of the models. Before that, here are the basic terms needed to understand-

True Positives (TP) - The correctly predicted positive values which means that the value of the actual class and the predicted class is true.

True Negatives (TN) - The correctly predicted negative i.e. wrong values which means that the value of the actual class and predicted class is false.

False Positives (FP) – When the actual class is false, and the predicted class is true.

False Negatives (FN) – When actual class is true but predicted class in false.

Once we understand and learn these four parameters, we can calculate Accuracy, Precision, Recall, and F1 score.

Accuracy - Accuracy is the most accurate performance measure, and it is a ratio of correctly predicted observations to the total observations.

Accuracy = TP+TN/TP+FP+FN+TN

Precision - Precision is the ratio of accurately predicted positive to the total predicted positive observations.

Precision = TP/TP+FP

Recall (Sensitivity) - Recall is the ratio of accurately predicted positive observations to all observations in the actual class.

Recall = TP/TP+FN

F1 score - It is calculated using Precision and Recall. Therefore, the F1 score takes both false positives and false negatives into account. F1-score works best if false positives and false negatives have similar costs.

$$F1\ score = 2 * \frac{Recall * Precision}{Recall + Precision}$$

The above measures were used on the model discussed in this section to measure the performance of the CNN models implemented.

We implemented two different CNN model architectures. The first proposed CNN architecture uses 32, 64, 64, 128-bit filter layers, respectively, with window size 3x3, activation function ReLu (Rectified Linear Unit), and Max pooling for each layer. The classification stage is implemented with dense/fully connected layers followed by the activation function SoftMax. The second model architecture works with 16,32,64,64–bit filter layers for each layer designed, and the rest of the architecture is similar to the above model. The first model was more accurate in the first attempt; hence, a further system was developed using that model architecture.

CHALLENGES

The Hand gesture recognition project, while promising in enhancing human-computer interaction, comes with its set of challenges. These are a few challenges faced while recognizing hand gestures:

1) Variability and complexity of hand gestures:
   The diversity in how people perform hand gestures poses a challenge for hand gesture recognition systems. Individuals exhibit variability in how fast they perform gestures, the orientations of their hand movements, and the specific stylistic nuances in their gestures. For example, a person may execute a waving gesture at a slow or rapid pace, change the angle or orientation of their hand while gesturing, or incorporate personal variations in how they express certain gestures—tried to attain approximate solutions by using diverse training data. Ensured the training dataset included diverse individuals performing gestures at different paces, orientations, styles, and backgrounds. This helped the model learn to recognize the inherent variability. Data augmentation techniques were used to introduce slight variations in the training dataset.

2) Lighting conditions and background noise can also impact recognition accuracy, making it challenging to design a robust system. Changes in lighting, such as low light or harsh shadows, can affect the quality and visibility of the captured images or video frames. Inconsistent lighting may result in poor contrast, making it difficult for the system to distinguish between different hand or track movement parts. Background noise, which includes irrelevant visual information or unrelated environmental movements, can interfere with accurately detecting and interpreting hand gestures. This interference may lead to misclassification or misinterpretation of gestures.

   One way to avoid background noises is to threshold the image so that only the hand is visible and the background is black. There are different thresholding techniques. Also can employ background subtraction techniques or background modeling to isolate the hand from the background

FUTURE SCOPE

The potential expansion of this project is extensive, given the ongoing evolution of research in sign language systems. Currently limited to American Sign Language, the model is anticipated to undergo enhancements enabling its functionality with Indian Sign Language. There is a need for further development in capturing dynamic gestures, as the current model is restricted to predicting static finger spellings. Improvements in datasets are essential, incorporating better and more accurate images, possibly by introducing variations in light density. Additional training aims to enable efficient detection of two-hand gestures. Sign language encompasses more than just alphabets and digits, including words and expressions.

The project also envisions the creation of an Android application designed for universal use, allowing communication with deaf and mute individuals proficient in sign language. Deploying the model into an application aims to bridge the communication gap between the hearing-impaired and the general population. The ongoing development focuses on creating an efficient and user-friendly app. The application will feature a module converting live-captured images to text and vice versa, recognizing the importance of understanding words and expressions in communication. Continuous training of the model is crucial to effectively detecting words and common expressions.

CONCLUSION

Our project aims to narrow the communication gap by introducing a cost-effective computer application. The goal is to automatically capture, recognize, and translate sign language into text, benefiting individuals who are deaf. The obtained image undergoes analysis, processing, and conversion, ultimately displayed either as sign language or text on the screen for the hearing impaired. Our research has demonstrated the effectiveness of Convolutional Neural Networks (CNN) in learning to identify and predict text. We have developed a model that preprocesses the image to meet the necessary requirements for input into the CNN. This system offers an approach to alleviate challenges in communicating with individuals facing speech disabilities. The CNN architecture proposed resulted in minimal training and validation loss. We have explored various image processing techniques to identify the most suitable one for our specific needs.

References

[1] Teena Varma, Ricketa Baptista, Daksha Chithirai Pandi, Ryland Coutinho, “Sign Language Detection using Image Processing and Deep Learning”, International Research Journal of Engineering and Technology (IRJET), vol. 7, issue 11, no. 55, Nov 2020

[2] Dr. M. Madhiarasan, Prof. Partha Pratim Roy, “A Comprehensive Review of Sign Language Recognition: Different Types, Modalities, and Datasets”, arXiv:2204.03328, 7 Apr 2022

[3] Ankita Wadhawan, Parteek Kumar, “Sign Language Recognition Systems: A Decade Systematic Literature Review”, Archives of Computational Methods in Engineering 28, 785–813, December 2019

[4] N. Mohamed, M. B. Mustafa and N. Jomhari, "A Review of the Hand Gesture Recognition System: Current Progress and Future Directions," in IEEE Access, vol. 9, pp. 157422-157436, 2021, doi: 10.1109/ACCESS.2021.3129650.

[5] H. Cooper, B. Holt, and R. Bowden, ‘‘Sign language recognition,’’ in Visual Analysis of Humans. London, U.K.: Springer, 2011, pp. 539–562

[6] Pankajakshan, P. C., & Thilagavathi B.. Sign language recognition system. International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), (2015).

[7] M. Mohandes, M. Deriche, U. Johar, and S. Ilyas, ‘‘A signer-independent Arabic sign language recognition system using face detection, geometric features, and a hidden Markov model,’’ Comput. Electr. Eng., vol. 38, no. 2, pp. 422–433, 2012.

[8] Das, A., Gawde, S., Suratwala, K., Kalbande, D. “Sign Language Recognition Using Deep Learning on Custom Processed Static Gesture Images”. 2018 International Conference on Smart City and Emerging Technology (ICSCET), (2018).

[9] Rao, G. A., Syamala, K., Kishore, P. V. V., Sastry, A. S. C. S. “Deep convolutional neural networks for sign language recognition”, (2018).

[10] Mahesh Kumar NB. "Conversion of Sign Language into Text". International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 9, (2018).

[11] Mediapipe hand landmark detection guide and models for Hand detection https://developers.google.com/mediapipe/solutions/vision/hand_landmarker

[12] Face and Hand Landmarks Detection using Python – Media pipe, OpenCV https://www.geeksforgeeks.org/face-and-hand-landmarks-detection-using-python-mediapipe-opencv/

[13] M. J. Cheok, Z. Omar, and M. H. Jaward, ‘‘A review of hand gesture and sign language recognition techniques,’’ Int. J. Mach. Learn. Cybern., vol. 10, no. 1, pp. 131–153, Jan. 2017, doi: 10.1007/s13042-017-0705-5

[14] B. K. Chakraborty, D. Sarma, M. K. Bhuyan, and K. F. MacDorman, ‘‘Review of constraints on vision-based gesture recognition for human– computer interaction,’’ IET Comput. Vis., vol. 12, no. 1, pp. 3–15, Feb. 2018, doi: 10.1049/iet-cvi.2017.0052.

[15] S. S. Rautaray and A. Agrawal, ‘‘Vision based hand gesture recognition for human computer interaction: A survey,’’ Artif. Intell. Rev., vol. 43, no. 1, pp. 1–54, Jan. 2012, doi: 10.1007/s10462-012-9356-9.

[16] M. A. Moni and A. B. M. S. Ali, ‘‘HMM based hand gesture recognition: A review on techniques and approaches,’’ in Proc. 2nd IEEE Int. Conf. Comput. Sci. Inf. Technol., 2009, pp. 433–437

[17] N. Aloysius and M. Geetha, ‘‘Understanding vision-based continuous sign language recognition,’’ Multimedia Tools Appl., vol. 79, nos. 31–32, pp. 22177–22209, Aug. 2020, doi: 10.1007/s11042-020-08961-z

[18] Kumar, A., Thankachan, K., & Dominic, M. M. “Sign language recognition”. 3rd International Conference on Recent Advances in Information Technology (RAIT), (2016).