International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 08 | August - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

Sign Language Recognition and Conversion to Audio using Image Processing and Machine Learning

Pragathi H[1], Roopa M S [2]

[1] (PG Scholar, Computer Science and Engineering Department, Dayananda Sagar College of Engineering, Bengaluru)
[2](Associate Professor ,Computer Science and Engineering Department, Dayananda Sagar College of Engineering, Bengaluru)

Abstract - Sign language is one of the oldest and most natural forms of language for communication. Sign language is used by speech-impaired persons for communication. For a regular individual, sign language interpretation is exceedingly challenging. This project effort proposes a real-time method for finger spelling-based American sign language using neural networks. The hand is first put through a Gaussian filter in this manner to get rid of the noise. A classifier is used to predict the class of the hand gestures from the filter's output. It displays the textformatted output for a specific sign from that class. Python's pyttsx3 module is used to convert text to speech.

Key Words: American sign language, neural networks,Gaussian filter,pyttsx3

1.INTRODUCTION

In a country as vast and diverse as ours, the array of official and regional languages presents a unique set of challenges. Maintaining the individuality and accuracy of language interpretation becomes particularly complex due to the sheer linguistic variety. These languages encounter difficulties when attempting to convey messages across various regions, civilizations, and circumstances.Among these languages, American Sign Language (ASL) holds significant importance, serving as an integral means of communication for the Deaf-mute community. This language is the lifeblood for individuals who are unable to hear or speak. It facilitates two-way interaction, catering to both those who are deaf or hard of hearing, as well as those who can hear but lack the ability to vocalize.Our primary objective in implementing this approach is to bridge the communication divide that separates the deaf and hearing communities. By doing so, we aim to dismantle the barriers that have hindered effective interaction between these two groups.

A. Overview of Sign Language

Sign languages exhibit numerous parallels with spoken languages, sharing fundamental traits that define them as authentic languages. They naturally emerge wherever there's a community of individuals engaged in communication. In essence, sign languages are on par with spoken languages in terms of effectively fulfilling the diverse social and cognitive roles that languages typically do. Children, through ordinary exposure and interaction, instinctively acquire sign languages in a manner akin to spoken languages, without formal instruction.However, it's important to recognize that sign languages possess their own distinct attributes. Being manual-visual languages, they harness an entirely different physical medium compared to the vocal-auditory system utilized by spoken languages. The utilization of these two distinct physical modalities—manual-visual and vocal-auditory—inevitably shapes the structure and composition of the languages they convey. This variance in physical representation is likely to contribute to notable dissimilarities in the way these languages are formed and transmitted.

B. Importance of Sign Language recognition

The inability to communicate verbally is widely recognized as a significant disability. However, individuals facing this condition have various avenues for communication, among which sign language stands out as one of the most prevalent methods. Sign language leverages the expressive potential of the human body, allowing effective communication through a series of distinct gestures. Each word in this language corresponds to a specific set of actions, forming a unique expression.An innovative approach aims to convert human sign language into audible speech using motion capture and gesture recognition technologies. This transformative process involves convolutional networks, a type of neural network particularly adept at addressing the challenges encountered by individuals who are unable to speak or hear. The recognition of these intricate gestures is a complex task, and it's a challenge that convolutional networks are equipped to tackle.Across various domains, efforts are being exerted to alleviate the struggles associated with communication impairments. Convolutional networks come to the forefront by employing distinct layers to interpret sign language, transforming it into both written text and audible speech. This technology plays a pivotal role in bridging the societal gap between those who possess verbal communication abilities and those who rely on sign language. Its capability to recognize and interpret sign language holds immense potential in fostering understanding, connection, and inclusivity within communities. By breaking down communication barriers for individuals with speech impairments, this technology contributes significantly to enhancing their quality of life.

C. Use of neural networks in sign language recognition

Convolutional Neural Networks (CNNs) serve as potent tools within the realm of deep learning, excelling particularly in image recognition tasks. Their design is rooted in the workings of our brain's visual system, drawing inspiration from the visual cortex's mechanisms. To envision this, consider the brain's visual cortex as a model for CNNs' operations.Within these networks, artificial neurons emulate a concept similar to focusing on specific areas within an image, resembling small, focused spots. These spots are referred to as receptive fields. The learning process of these neurons involves employing diverse filter values, akin to adjustable weights, to analyze these focused image sections. For each part of the image, multiple filters are applied, generating specialized maps that highlight significant features. These vital features are determined through a procedure termed activation.Following this, the salient aspects extracted from these maps are consolidated using a technique known as pooling. These steps transpire across various layers, resulting in CNNs' exceptional prowess in comprehending images. Through this intricate interplay of convolution, activation, and pooling, CNNs excel in recognizing patterns, structures, and characteristics in visual data, mirroring the way our brain processes visual information.

D. Advantage of text to audio conversion of sign language

Transforming sign language text into audio format represents a remarkable stride in enhancing accessibility for the speech-impaired community. This technological breakthrough serves to bridge the communication divide by affording individuals who depend on sign language the ability to grasp spoken language through the generated audio output. What's more, this innovation extends its impact to include those who are illiterate. The translated sign language, now rendered audibly, becomes accessible to individuals who encounter difficulties in reading.By catering to the needs of both the speech-impaired and illiterate populations, this advancement champions inclusivity. It ensures that vital messages conveyed through sign language can be comprehended and embraced universally. This novel approach not only fosters understanding but also facilitates a sense of connection and unity across different communication styles and abilities. In essence, it contributes significantly to breaking down barriers and promoting a more inclusive society.

E. Role of pyttsx3 in sign language

The Pyttsx3 library is used to change text into spoken words. Think of it as a tool that takes written words and turns them into speech. It's like a special translator that works without needing the internet, unlike some other similar tools. When the library knows which letter is predicted, it takes that letter and makes it into sound, like saying "A" out loud. So, the library helps change the computer's understanding of letters into something you can hear.

F Organisation of Paper

In section (2) takes a more in-depth with a survey on existing literature available.
Section (3) looks at the architecture of the proposed solution with an overview of the system design
Section (4) dives into the Implementation of the solution
Section (5) describes results and discussion
Section (6) summarizes our findings and conclusion

2. RELATED WORKS

Rajarshi Bhadra and Subhajit Kar propose a deep, multi-layered Convolutional Neural Network (CNN) for the automatic detection and classification of sign languages from hand gesture images. The system aims to assist people with hearing and speech disorders by accurately recognizing and interpreting sign language. The proposed CNN architecture consists of five layers, including convolutional and max-pooling layers. It uses both static (alphabets and digits) and dynamic (emotional states) hand gesture images in the training and validation phases to enhance robustness. The experimental results demonstrate the effectiveness of the proposed methodology, achieving a blind test accuracy of 99.89%. However, real-time detection and classification are not implemented in this study, leaving room for future research in developing intelligent methodologies for real-time sign language detection. Overall, the proposed system shows promising potential for overcoming communication barriers and facilitating better communication for individuals with hearing and speech impairments [1].

Kanchan Dabre and Surekha Dholay propose a marker-free, visual Indian Sign Language (ISL) recognition system using image processing, computer vision, and neural network methodologies. The system aims to identify hand gestures in images taken from a video through a webcam, convert them into text, and then further convert them into audio to aid communication for hearing-impaired individuals. The system consists of two stages: preprocessing (image processing) and classification (using the Haar Cascade Classifier). The preprocessing phase involves extracting hand shapes and features from the input video frames, while the classification phase uses the Haar Cascade Classifier to classify the signs based on the learned features. The system achieves an average accuracy rate of 92.68% for interpreting ISL signs. Future scope includes expanding the vocabulary, incorporating facial expressions and whole-body gestures, and optimizing speech synthesis for faster responses. The system's potential applications extend to sign language education and human-computer interaction for hearing-impaired individuals [2].

Zhibo Wang et al. introduce DeepSLR, a real-time end-to-end sign language recognition (SLR) system aimed at facilitating communication between hearing-impaired individuals and the general population. Existing SLR systems have limitations, either lacking continuous recognition or suffering from low accuracy due to sign segmentation difficulties and insufficient capture of finger and arm motions. DeepSLR addresses these challenges through a novel approach utilizing two armbands equipped with IMUs and multi-channel sEMG sensors to capture both coarse-grained arm movements and fine-grained finger motions. The system adopts an attention-based encoder-decoder model with a multi-channel CNN, enabling accurate, scalable, and end-to-end continuous SLR without sign segmentation. Implemented on a smartphone, DeepSLR achieves real-time recognition, making it practical for everyday use. The paper presents a dataset of Chinese Sign Language (CSL) with 60 sentences and 20,000 samples from 34 volunteers. DeepSLR demonstrates an impressive average word error rate (WER) of 10.8% for continuous sentence recognition and efficiently recognizes a four-word sentence in less than 1.1 seconds [3].

Ankit Ojha et al. propose a desktop application that utilizes computer vision and neural networks to facilitate real-time translation of American Sign Language (ASL) gestures. With the help of a computer's webcam, the application captures the hand gestures made while signing ASL and processes them using a Convolutional Neural Network (CNN). CNNs are known for their effectiveness in handling computer vision tasks, and in this case, they efficiently recognize and identify the different ASL gestures with a high level of accuracy. Once the gestures are recognized, the application converts them into corresponding text, which is then further transformed into speech, allowing for seamless communication between the hearing-impaired and the hearing community. The system's success lies in its ability to bridge the communication gap, enabling deaf individuals to interact effectively with those who do not know sign language. Though the current version focuses on finger spelling translation in ASL, there is potential to expand its capabilities to other sign languages with sufficient training data [6].

Raimundo F. Pinto Jr. et al. introduce a methodology for recognizing static gestures in American Sign Language (ASL) using convolutional neural networks (CNNs). The motivation for this research is the ability of humans to recognize the body and sign language through a combination of vision and synaptic interactions in the brain. To achieve this on computers, various challenges need to be addressed, such as object separation in images and appropriate image capture technology and classification techniques. The development of devices like Kinect and Leap Motion has enabled human-machine interaction through gesture capture. The gesture recognition methods are divided into static and dynamic categories, with static gestures involving single-image processing and dynamic gestures requiring image sequences. Several supervised and unsupervised learning methods have been used for gesture recognition, including neural networks, convolutional neural networks, support vector machines, and others. The proposed methodology utilizes color segmentation, morphological operations, and polygon approximation to preprocess the input images. A skin color segmentation technique using an MLP neural network is employed to separate the hand region from the background. After preprocessing, the images are fed into various CNN architectures for classification. The experiments are conducted on two image datasets: a self-acquired dataset of ASL static gestures and a dataset available in the literature. The CNNs are trained and tested, and results are compared with other existing networks and recognition methodologies. The approach shows promising results for static gesture classification, achieving excellent performance. The experiments are performed using cross-validation, and metrics such as accuracy, recall, and F1 score are used to evaluate the models [7].

3. SYSTEM DESIGN

This section introduces an architecture for sign language translation and audio recognition using machine learning and image processing.

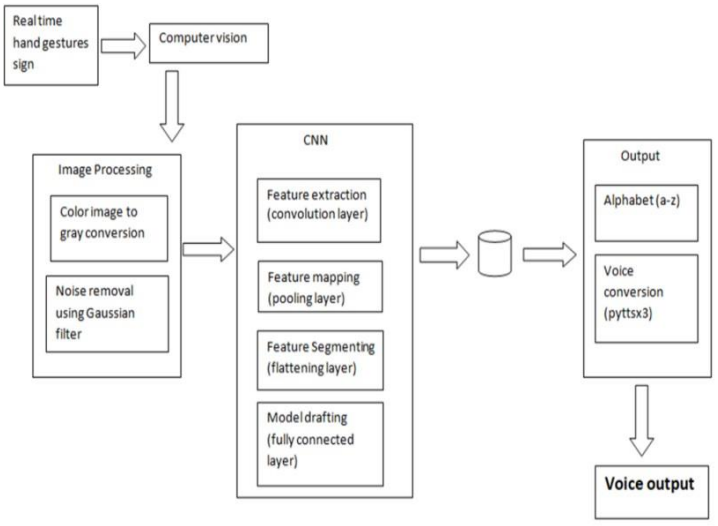

Fig -1: System Architecture of Proposed Method

The proposed system aims to break down communication barriers faced by deaf and dumb individuals by offering a two-way communication solution. The system takes input in the form of hand gestures made by the users and converts them into commonly used words for communication. Once the gestures are processed and converted into text, the system further transforms the text into voice. This voice output allows normal people to comprehend the message conveyed by deaf

and dumb individuals. Likewise, the system enables normal people to speak in their voices, which is then converted into text and translated into actions that can be easily understood by deaf users.In conclusion, the technology bridges the communication gap between deaf and dumb people and hearing people by translating gestures into text and text into voice for mutual understanding.

4. PROPOSED METHODOLOGY

A. Module 1 - Dataset Creation

1) Image capturing:

In this paper, the dataset creation process involves utilizing a video camera to stream live footage. Each frame of the video that contains a hand within the Region of Interest (ROI) is saved in a directory, which is further divided into "train" and "test" folders. Within these folders, there are 26 sub-folders representing different classes or labels of images that were collected during the process.To access the live camera feed and set up the ROI for hand recognition, the project uses the OpenCV library. OpenCV provides functionality to capture video frames and process them in real-time. By defining the ROI, the system focuses only on the relevant part of the frame where hand movements need to be recognized.Once the ROI is defined, the next step involves preprocessing the images. The RGB image of the ROI is first converted to a grayscale image. Grayscale images only contain intensity values, which simplifies the processing and reduces computational complexity. This conversion is beneficial for many computer vision tasks, including hand gesture recognition.After converting to grayscale, a Gaussian blur filter is applied to the image. Gaussian blur is a smoothing technique that helps reduce noise and sharp edges, making it easier to extract relevant features from the image. The blur filter assists in removing unwanted details that could interfere with hand motion recognition.Next, the image undergoes a binary thresholding process. By using the thresholding approach, grayscale images can be turned into binary images, where each pixel is either black or white depending on the threshold value. This binary transformation simplifies the image even further and emphasizes the hand's outline and key features.By combining the aforementioned steps, the final result is a dataset containing binary images of hands in various positions and gestures, collected from the live camera feed. These images are organized into separate folders based on their classes or labels, providing the necessary data for training and testing the hand gesture recognition model.

2) Image Augmentation:

Image augmentation is a valuable technique, particularly when dealing with datasets that have a limited number of samples. It helps to artificially increase the diversity and size of the dataset by applying various transformations to the existing images. This is especially important for training machine learning models, as a larger and more diverse dataset can improve the model's generalization and performance on unseen data.Three common image augmentation parameters frequently used to augment the dataset are zoom, shear, and rotation. These parameters enable the generation of additional variations of the original images, thereby enriching the dataset and making it more representative of real-world scenarios.The zoom augmentation parameter allows for enlarging or shrinking the image based on a defined range of pixel values. By randomly zooming in or out of the original images, different perspectives and scales of the hand gestures can be created, making the model more robust to variations in hand size and distance from the camera.The shear range parameter introduces shearing transformations to the images. Shearing involves displacing points in the image along a particular direction, effectively slanting or skewing the image. By applying shearing transformations with a specified range, the dataset can include images of hands in various orientations and angles, improving the model's ability to recognize gestures from different viewpoints.Lastly, the horizontal fitment augmentation parameter enables flipping the images horizontally, effectively reversing the active layer from left to right. This augmentation technique introduces mirror images of the original hand gestures, further diversifying the dataset and accounting for scenarios where hands might be seen from opposite directions.By employing these image augmentation techniques, the dataset can be significantly expanded with new and diverse samples. As a result, the trained model is better equipped to handle different hand gestures, sizes, orientations, and lighting conditions, leading to improved performance and accuracy in real-world applications.

B. Module 2 - CNN Model

In this paper, a Convolutional Neural Network (CNN) architecture is designed to recognize hand gestures represented as images. The input images have a resolution of 64x64 pixels. The CNN consists of multiple layers, starting with convolutional and pooling layers for feature extraction, followed by densely connected layers for classification.The first convolution layer processes the input images using 32 filter weights to extract low-level features from the images. Subsequently, a maximum pooling layer with a pool size of 2x2 downsamples the feature maps, reducing their spatial dimensions.The second convolution layer accepts the output of the first pooling layer as input and employs 64 filter weights to extract more complex features from the downsampled feature maps. Another maximum pooling layer with a pool size of 2x2 further reduces the spatial dimensions.The third convolution layer takes the output of the second pooling layer as input and utilizes 128 filter weights to extract even higher-level features from the downsampled feature maps.The flattened output from the third convolution layer is passed through the first densely connected layer, which consists of 128 neurons. To prevent overfitting, a dropout layer with a dropout rate of 0.2 is applied, randomly

deactivating 20% of the neurons during training.The output of the first densely connected layer is then fed to the second densely connected layer, also containing 128 neurons, with another dropout layer to avoid overfitting.Next, the output from the second densely connected layer is used as input for the third densely connected layer, which contains 64 neurons.The final layer, also known as the output layer, has the same number of neurons as the number of classes being classified, which in this case are the different alphabets for hand gestures. This layer produces the final predictions for each input image. To load and preprocess the data, the Image Data Generator of Keras is used.The flow_from_directory function is employed to load the training and test sets of data.Once the model is defined, it is fitted to the train batches for 10 epochs by default, but the number of epochs can be adjusted. After each epoch, the accuracy and loss metrics are computed using the validation dataset, providing insights into the model's performance during training.Finally, the model is saved for later use in the last module of the project. This CNN architecture, along with the image preprocessing and data loading steps, facilitates the recognition of hand gestures from images, making it suitable for various applications, such as sign language translation or human-computer interaction systems.

1) Predict the gesture:

The paper next phase involves real-time hand gesture recognition using the trained model. The process unfolds as follows:

To detect the Region of Interest (ROI) in the live camera feed, a bounding box is created around the area where the hand gestures are expected to appear. This bounding box helps isolate the hand from the rest of the scene, focusing the analysis only on the relevant region.The trained model, previously saved during the training phase, is now loaded using Keras' models.load_model function. This restores the CNN model with its learned parameters and weights, making it ready for predictions.As the live camera feed continuously streams frames, the system detects and localizes the hand within the defined ROI using the bounding box. The extracted ROI, containing the hand gesture, is then fed into the loaded model for prediction.Using the trained CNN model, the system predicts the alphabet corresponding to the detected hand gesture in real-time. The model processes the ROI image and classifies it into the appropriate letter of the alphabet, based on its learned patterns and features.Finally, the predicted alphabet representing the recognized hand gesture is displayed on the output screen. This feedback provides users with real-time information about the detected hand gesture.

C. Module 3 - Conversion of text to audio

For the purpose of converting text to speech, the paper utilizes the Pyttsx3 library. This library serves as the tool responsible for the conversion process. When the classifier predicts a specific alphabet text corresponding to a hand gesture, this predicted text is passed as input to the pyttsx3 library. The primary function of the pyttsx3 module is to transform this text into audible speech. Notably, this process doesn't require an internet connection, setting it apart from other available libraries that might depend on online resources.

D. Module 4- Accuarcy

Calculating accuracy for sign language recognition involves comparing the model's predictions to the actual labels (expected outputs) and determining how many predictions were correct.
Here's the basic formula:

Accuracy = (Number of Correct Predictions) / (Total Number of Predictions) * 100

Collect Data: You need a dataset with labeled examples of sign language gestures. Each example should have an image of the gesture along with its corresponding label.
Train the Model: Train a convolutional neural network (CNN) using your dataset. The network learns to associate visual features of gestures with their respective labels.
Validation Data: Set aside a portion of your dataset for validation. This data is not used during training but helps you assess the model's performance.
Prediction and Comparison: Use the trained model to predict labels for the validation set. Compare these predictions with the actual labels.
Calculate Accuracy: Count the number of predictions that match the actual labels. Divide this by the total number of predictions.

Algorithm

Step 1: Input Hand Gesture - Users perform hand gestures in front of the camera.
Step 2: Image Capturing using OpenCV - Capture live video frames using the OpenCV library.
Step 3: Image Processing - Define a Region of Interest (ROI) around the hand gesture. - Convert the ROI image to grayscale. - Use Gaussian blur to reduce noise.
Step 4: Feature Extraction - Apply binary thresholding to create a binary image. - Prepare the processed image for CNN input.
Step 5: Model Generation - Design a CNN model architecture. - Configure convolutional layers, pooling layers, and fully connected layers. - Compile the model with appropriate loss and optimization functions.
Step 6: Sign Recognition - Prepare labeled datasets of hand gesture images. - Split data into training and validation sets. - Using the training set of data, train the CNN model. - Make use of the validation set to test the model.
Step 7: Text Generation - Capture new video frames with hand gestures. - Preprocess the frames as done during training. - Feed the preprocessed frames to the trained CNN model. - Obtain the predicted hand gesture (text representation).
Step 8: Voice Conversion using Pyttsx - Use the Pyttsx library for text-to-speech conversion. - Convert the predicted hand gesture (text) into audio. - Output the audio to provide audible recognition.

End of Algorithm.

5. IMPLEMENTATION

A .Image Processing module

The initial step of the paper involves capturing images using OpenCV and a computer webcam. The red, green, and blue color channels of each pixel are represented by these images, which are commonly in the RGB format.After capturing the RGB images, the next step is to convert them into grayscale images. Grayscale images contain only intensity values, representing the brightness or darkness of each pixel. This conversion simplifies the images and reduces their dimensionality, making them easier to process.Once the images are converted to grayscale, a Gaussian blur filter is applied to remove background noise and smooth out the images. Gaussian blur is a popular image filtering technique that helps reduce noise and unwanted details while preserving the essential features of hand gestures.After applying the Gaussian blur, a binary thresholding operation is performed on the images. By applying a predetermined threshold value, thresholding turns the grayscale images into binary images, where each pixel is categorized as either black or white. This process simplifies the images even further, enhancing the contrast and emphasizing the hand gestures' outlines.The final outcome of this module is a collection of processed images that have been grayscaled, blurred with a Gaussian blur effect, and thresholded into binary pictures. These processed images are now ready to be fed into the Convolutional Neural Network (CNN) for training or used as input for hand gesture recognition tasks.

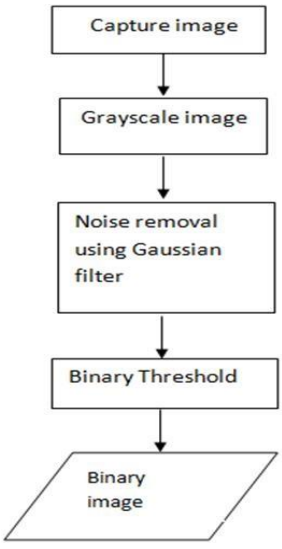

Fig -2: Flow chart of proposed method module A

B. CNN Module

In the first module of the paper, the input images of hand gestures are preprocessed and prepared for training the Convolutional Neural Network (CNN). These processed images are then used as training data for the CNN module.In the CNN module, multiple convolutional and pooling layers are added to the model to extract relevant features from the input images.The convolutional layers use filters to the images to recognize different patterns and traits, while the pooling layers downsample the feature maps. This reduces the spatial dimensions and computational complexity.After passing through the convolutional and pooling layers, the feature maps are flattened into a 1D array using the Flatten layer. This flattening process reshapes the multi-dimensional feature maps into a single vector, which can be fed as input to the fully connected layers.The flattened array is then passed through a fully connected layer containing a number of neurons. This layer learns to associate the extracted features with the corresponding hand gestures' classes (letters of the alphabet). The fully connected layer uses these learned associations to make predictions for new and unseen hand gesture images.The output of this module is the trained model, which includes all the learned parameters and weights of the CNN. This trained model is capable of recognizing hand gestures and classifying them into the respective alphabet classes.

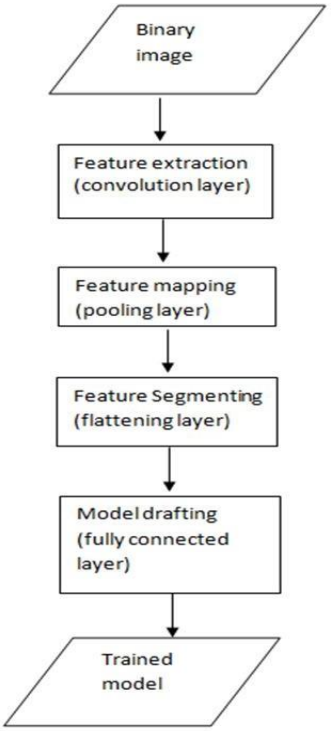

Fig -3: Flowchart of proposed method module B

C. Audio Conversion Module

In this paper, the trained model is utilized for real-time hand gesture prediction. Once the output screen is open, the system awaits hand gestures as input from the user. As the

user performs a hand gesture in front of the camera, the system captures the corresponding image of the gesture.Using the loaded and trained model, the captured image of the hand gesture is passed through the model for prediction. The model has been specifically designed to predict alphabet gestures, meaning it classifies the hand gestures into different letters of the alphabet.After the prediction is made, the system converts the predicted letter into audio using the pyttsx3 library of Python. The pyttsx3 library is capable of converting text into speech, allowing the system to audibly announce the recognized letter to the user.

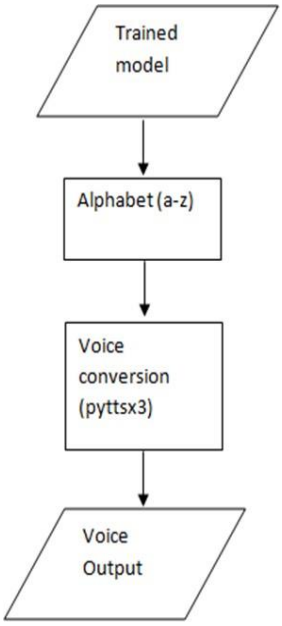

Fig -4: Flowchart of proposed method module C

6. RESULTS AND DISCUSSION

A. Data collection screen

1) Without hand

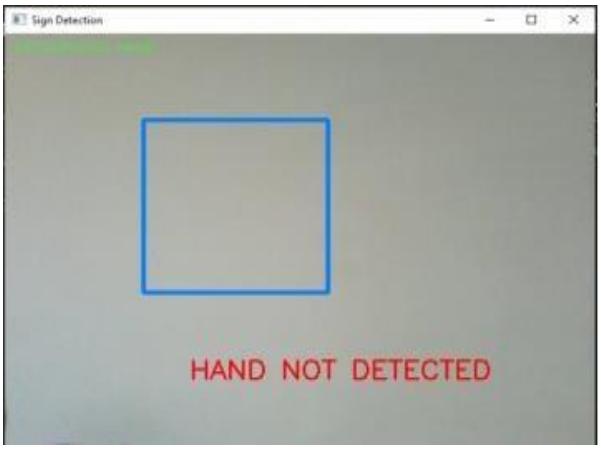

Fig -5: Without Hand

Fig 5 above shows a screen with ROI where our hand gesture is not recognized.

2) With Hand

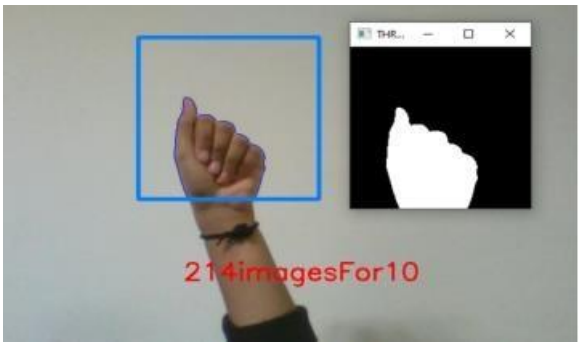

Fig -6:.With Hand

Fig 6 above shows a screen with ROI where our hand gesture is recognized.

3) Alphabet Predicted

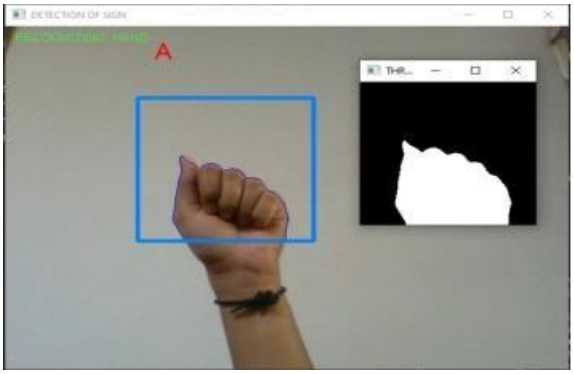

Fig -7:.Alphabet Predicted

Fig 7 above shows a screen where hand gesture is detected with alphabet A is predicted.

7. CONCLUSION

In this paper, real-time vision-based American sign language recognition for deaf and dumb people has been developed for ASL alphabets. We can detect almost all the symbols provided that they are shown properly, there is no noise in the background, and the lighting is adequate. The given image of the sign is first converted into a binary image and predicted using a trained model. The predicted sign is converted to the alphabet and then to audio. The final output of the model is audio. The overall accuracy of the model is 85.7%. The accuracy of each alphabet is also calculated. The future scope of the proposed system includes enhancing model accuracy through more extensive data and advanced algorithms. Developing a dedicated application will improve

the user experience. Training the model for numbers, words, and sentences will enable comprehensive communication. Exploring alternative APIs for audio synthesis will offer diverse voice options for users' preferences and accessibility.

REFERENCES

[1] Rajarshi Bhadra and Subhajit Kar ,"Sign Language Detection from Hand Gesture Images using Deep Multi-layered Convolution Neural Network" , 2021

[2] Kanchan Dabre and Surekha Dholay , "Machine Learning Model for Sign Language Interpretation using Webcam Images",2014

[3] Zhibo Wang, Senio,Tengda Zhao, Jinxin Ma, Hongkai Chen, Kaixin Liu, Huajie Shao, Qian Wang, Ju Ren,''Hear Sign Language: A Real-time End-to-End Sign Language Recognition System'',2020

[4] Muhammad al-qurishIi , Thariq Khalid, and Riad Souissi "Deep Learning for Sign Language Recognition: Current Techniques, Benchmarks, and Open Issues",2021

[5] Boban Joksimoski, Eftim Zdravevski , Petre Lameski, Ivan Miguel Pires , Francisco José Melero , Tomás Puebla Martinez, Nuno M. Garcia , Martin Mihajlov, Ivan Chorbev , and Vladimir Trajkovik, ''Technological Solutions for Sign Language Recognition: A Scoping Review of Research Trends, Challenges, and Opportunities'' ,2022

[6] Ankit Ojha, Ayush Pandey, Shubham Mourya, Abhishek Thakur, Dr. Dayananda P, Sign Language to Text and Speech Translation in Real Time Using Convolutional Neural Network, International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 NCAIT - 2020 Conference Proceedings.

[7] Raimundo F. Pinto Jr. , Carlos D. B. Borges, Antoˆnio M. A. Almeida,Iális C. Paula Jr."Static Hand Gesture Recognition Based on Convolutional Neural Networks",Hindawi Journal of Electrical and Computer Engineering Volume 2019 Article ID 4167890.

[8] Rishin Tiwari, Saloni Birthare , Mr. Mayank Lovanshi, "Audio to Sign Language Converter",2022

[9] Prof. Abhishek Mehta, Dr. Kamini Solanki2 , Prof. Trupti Rathod ''Automatic Translate Real-Time Voice to Sign Language Conversion for Deaf and Dumb People'' 2021

[10] Paulraj M P, Sazali Yaacob ,Hazry Desa, Hema C.R. ,M. Hariharan, Wan Mohd Ridzuan ,Wan Ab Majid , "Sign Language Into Voice Signal Conversion Using Head And Hand Gestures",2008

[11] https://github.com/luvk1412/Sign-Language-To-Text

[12] https://en.Wikipedia.org/wiki/Tensorflow

[13] https://pypi.org/project/pyttsx3

[14] N. H. Dardas and N. D. Georganas, "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques," IEEE Transactions on Instrumentation and Measurement, vol. 60, no. 11, pp. 3592–3607, 2011.

[15] G. A. Rao, K. Syamala, P. V. V. Kishore, and A. S. C. S. Sastry, "Deep convolutional neural networks for sign language recognition," in 2018 Conference on Signal Processing And Communication Engineering Systems (SPACES), 2018, pp. 194–197.

[16] M. R. Abid, E. M. Petriu, and E. Amjadian, "Dynamic sign language recognition for smart home interactive application using stochastic linear formal grammar," IEEE Transactions on Instrumentation and Measurement, vol. 64, no. 3, pp. 596–605, 2015.

[17] D. Avola, M. Bernardi, L. Cinque, G. L. Foresti, and C. Massaroni, "Exploiting recurrent neural networks and leap motion controller for the recognition of sign language and semaphoric hand gestures," IEEE Transactions on Multimedia, vol. 21, no. 1, pp. 234–245, 2019.

[18] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," arXiv preprint arXiv:1505.00853, 2015.

[19] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," arXiv preprint arXiv:1611.01144, 2016.

[20] L. Lu, Y. Shin, Y. Su, and G. E. Karniadakis, "Dying relu and initialization: Theory and numerical examples," arXiv preprint arXiv:1903.06733, 2019.