Gesture based Real-Time Sign Language Recognition System

1st Tiya Ann Siby
Department of Information Technology
School of Engineering & Technology
Christ (Deemed-to-be University)
Bengaluru, India
tiya.ann@btech.christuniversity.in

2nd Sonam Pal
Department of Information Technology
School of Engineering & Technology
Christ (Deemed-to-be University)
Bengaluru, India
sonam.pal@btech.christuniversity.in

3rd Jessica Arlina
Department of Information Technology
School of Engineering & Technology
Christ (Deemed-to-be University)
Bengaluru, India
jessica.arlina@btech.christuniversity.in

4th Shamanth Nagaraju
Department of Computer Science & Engineering
School of Engineering & Technology
Christ (Deemed-to-be University)
Bengaluru, India
shamanth.n@christuniversity.in

***Abstract*—Real-Time Sign Language Recognition (RTSLG) can help people express clearer thoughts, speak in shorter sentences, and be more expressive to use declarative language. Hand gestures provide a wealth of information that persons with disabilities can use to communicate in a fundamental way and to complement communication for others. Since the hand gesture information is based on movement sequences, accurately detecting hand gestures in real-time is difficult. Hearing-impaired persons have difficulty interacting with others, resulting in a communication gap. The only way for them to communicate their ideas and feelings is to use hand signals, which are not understood by many people. As a result, in recent days, the hand gesture detection system has gained prominence. In this paper, the proposed design is of a deep learning model using Python, TensorFlow, OpenCV and Histogram Equalization that can be accessed from the web browser. The proposed RTSLG system uses image detection, computer vision, and neural network methodologies i.e. Convolution Neural Network to recognise the characteristics of the hand in video filmed by a web camera. To enhance the details of the images, an image processing technique called Histogram Equalization is performed. The accuracy obtained by the proposed system is 87.8%. Once the gesture is recognized and text output is displayed, the proposed RTSLG system makes use of gTTS (Google Text-to-Speech) library in order to convert the displayed text to audio for assisting the communication of speech and hearing-impaired person.**

***Index Terms*—Sign Language Recognition, Hand Gesture Detection, Deep Learning, Convolutional Neural Network, Text-to-Sound Converter**

I. Introduction

Real-Time Sign Language Recognition (RTSLG) is a gesture based speaking system, especially for the deaf and dumb. Hand gesture recognition may be used for a variety of reasons. It may also be used as a translator, translating gestures into words, and for human-computer interface, making human-machine interaction easier [1]. To express messages, three-dimensional spaces and hand gestures, i.e. together with other parts of the body, are employed. It has a unique vocabulary and grammar that is unrelated to spoken or written languages. Sign language, unlike spoken conversation, makes use of visual skills. To construct comprehensive messages, spoken language employs rules; similarly, sign language is guided by a complicated grammar [2]. A sign language recognition system is a system that converts sign language into text or voice in a simple, efficient, and accurate manner. To detect the alphabet flow and translate sign language words and phrases, computerised digital image processing and a number of classification algorithms are applied. Hand gestures are an important aspect of nonverbal communication and play an important role in our everyday lives. There is no global sign language, and only a few individuals are familiar with it, making it an ineffective means of communication. Sign language recognition helps such people understand and communicate with people with disabilities without any misunderstanding.

This paper focuses on sign language recognition for both deaf and dumb people as it displays the text output for deaf people and produces audio output for the blind using the sign language. The motivation behind this work is to build a low-cost feasible system that is easily accessible by everyone. Since it is important to understand the basics of sign language in order to communicate, the proposed system has used the 26 alphabets of American Sign Language instead of using sentences, which would only work for the handful of chosen words and sentences. The proposed RT-SLG aims to eliminate the need for a human interpreter during communication and thus empowering deaf and dumb people. It is inspired by the need for someone to take up the challenge and respond to the responsibility of the society to help those in need.

The paper is organized as follows. Section II presents related work in the sign language recognition domain. The proposed RT-SLG is briefed in Section III. The results and performance analysis of the proposed RT-SLG are presented in Section IV. Conclusions and future work are summarized in Section V.

II. Literature Survey

In [3] automated approaches are presented. Identification of word limts in Japanese characters' continuous sentence phrases are performed in this paper. In the proposed technique a pair of joint segments are used to configure domain-based spatio-temporal angle and distance. A binary random forest model is also used for autonomous word synthesis from motion sequences. Authors Rashmi R. Koli and Tanveer I. Bagban stress the necessity of proper use of Deep learning for many applications of motion data [4]. The major goal of this paper is to create a system for hearing-impaired people. A hand gesture recognition software is used to identify and convert hand movements into their intended meaning or sentence. Deep learning-based approach for sign language gesture recognition with efficient hand and gesture representation is emphasised in [5]. The suggested method is tested on a difficult dataset that includes 40 dynamic hand signs made by 40 people in an uncontrolled setting. The results reveal that the suggested system exceeds current best practises, confirming its efficacy. The appropriate mix of the skeleton and video functionalities which is excellent for SLR tasks are proposed in [6]. This paper offers a deep learning-based framework. Each sign recorded in the video translates to a word. Video (pictures also optical flow) skeletal features (body, hands, face) contributes to the gesture recognition. Additionally, a number of fusion techniques are applied to select the optimal strategy, which is an approach that combines data from several functional representations and incorporates the SLR methodology. In [7] the significance of key frame extraction for sign language recognition is emphasised. The density and dynamics of the sign language trajectory are used to extract key frames. To acquire the ultimate result of dynamic sign language recognition, it is required to merge hand trace with key movement type information. For the identification of dynamic gestures, positional and geometric aspects of dynamic motions are retrieved over time. The authors in [8] demonstrate the human hand's gestures. The motion is recorded and evaluated. The suggested approach simplifies the description of human hand gestures and allows them to be implemented in real-time. It detects hand motions and identifies the sign language denoted by that motion

Use of Long Short Term Memory (LSTM) to translate Indian sign language in real-time is underlined in [9]. The paper shows how to design and use sensor embedded gloves for gesture recognition. The proposed system recognizes the hand gestures relevant to Indian Sign Language (ISL) characters and converts it into voice. The glove uses Bluetooth to communicate data from the inbuilt sensor to filtering unit, such as a PC or smartphone. For the matching text and audio output, this data is subsequently classified using LSTM, a basic neural network. The authors stress the need of using a multi-view hand skeleton to recognise sign language in [10]. The authors propose a new deep learning-based pipeline architecture with single shot detector (SSD), 2D convolutional neural network (2D CNN), 3D Convolutional Neural Network (3DCNN), and LSTM. RGB input video is used for fast automatic hand gesture identification. Key points of the hands are estimated using CNN based model using 2D input frames. Authors Siddharth S. Rautaray and Anupam Agrawal created a virtual game that recognises hand gestures [11]. Traditional input devices can interact with the virtual world using the proposed hand gesture detection system. The suggested gesture-based interaction interface falls within the categories of virtual reality, sign language, and gaming. An extensive study on several sign language recognition algorithms is performed in [12]. It emphasises on mapping unsegmented video streams to gloss, with a concentration on sign language detection. Two novel sequence training criteria from the region are used in this challenge. Text recognition for speech and scenes is added in this paper.

The relevance of real-time sign language identification based on video streams is demonstrated by authors in [13]. To boost performance, they provide a 3DCNN technique combined with optical flow processing. High-density optical flow collects and processes RGB video streams before feeding them to 3DCNN to extract accurately. The authors have shown the character implementation speech recognition approach based on Kinect in [14]. Kinect is a 3D somatosensory camera that was introduced by Microsoft. It consists of RGB cameras and two depth sensors. Color, depth, and a skeletal frame are all available in this approach.

The Region of Interest (ROI) selection approach is used in [15]. In terms of effectiveness and real-time recognition via streaming video via cameras, the procedure using the ROI selection strategy outperforms conventional methods. Moreover, this technique provides an effective design that allows for the easy inclusion of extra indicators to the finalized Raspberry Pi prototype. In [16], the model integrates an RGB camera with a Raspberry Pi, a common technology nowadays due to its consistent performance. The radian fingertip analysis approach is used to identify motions within every frame, which is a unique method that does not need data training. This approach is well-suited to light effects in a complicated setting.

After an extensive literature survey, it is found that some of these papers are about systems that only give output for numbers 1-10 or some of the alphabets [1] or only a few sentences [3]-[5], [8], [11]. In some papers, the output is only text and in some exceptions which provide voice output, is implemented via Raspberry Pi [15] and [16], which is harder to implement and for people to carry it around. In this paper, the main focus is to develop a real-time sign language recognizer for all 26 alphabets from A-Z with voice output using gTTS (Google Text-to-Speech) library. This proposed system is built using a combination of Convolution Neural Network and Histogram Equalization(CNN+HE).

III. Proposed Work

There is a wall between general people and people with hearing and speaking disabilities in the form of communication, or rather the lack of communication. Usually, people don't tend to learn sign languages unless it directly affects them or their loved ones. As a result, there is a lot of space for miscommunication. This could also result in people with disabilities missing out on expressing themselves or showcasing and contributing their talents to their respective field of work. Sign language is becoming the dominant form of engagement and education for deaf people. It would be extremely advantageous for deaf and mute individuals all around the world if a sign language detector could be built to work in real-time in the browser. Using the real-time video to recognize and classify sign language gestures is a challenging task which is the motivation for the proposed RTSLG system. Along with this, additional text to voice output feature enables the two-way communication for both deaf and dumb.

A. Proposed Real-Time Sign Language Recognizer

The proposed RTSLG system is composed of data collection, data pre-processing extraction, followed by defining the neutral network. The proposed RTSLG starts with data collection stage. First, the train image count should be around 1000. The images are collected by showing the signs or gestures to the camera and simultaneously press caps lock of that particular alphabet to capture the images. Next in validation, the same process is repeated for validation and will be capturing almost 500 images for validation of data. When the dataset is created, the dataset is further pre-processed with image processing techniques, that is used to remove the irrelevant images from the dataset. Histogram equalization is also done in this pre-processing step. Histogram equalization is performed which is a fine tuning method. In histogram equalization, all the images are converted to have similar properties like same size, color, sharpness, etc. This is done manually for all 26 alphabets. This process makes model training and reading the images much easier. Then, converted images are stored in an allotted folder. The proped RTSLG system performs a feature extraction technique after the data pre-processing, in order to eliminate the unnecessary information from the dataset. As a result, grey images are obtained after performing feature extraction which in turn contains only useful information for the training of the CNN+HE model. The grey images obtained assist in reducing the computational complexity of the the proposed RTSLG system in comparison to the raw colour image-based datasets.

The CNN+HE model used in the proposed system uses the extracted images in its training. During the training time, the CNN+HE model interprets the extracted images and the prediction is done accordingly. As a result, the CNN+HE model employed in the proposed approach, significantly reduces the prediction error . Real-time testing is when the predictions happens, where all the 26 Alphabets will be predicted in real-time. In evaluation, another ASL dataset from Kaggle [17] is used. The original dataset is validated with the other dataset and a confusion matrix with the results is plotted.

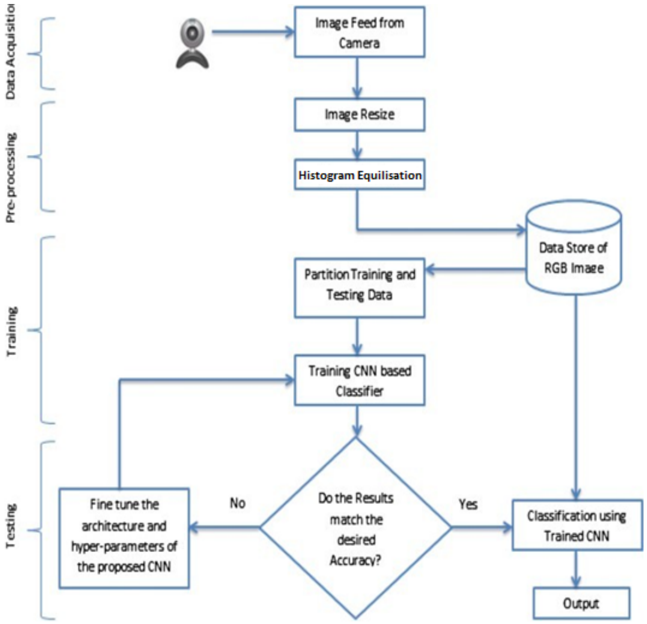

Data Acquisitio
Pre-processing
Training
Testing
Image Feed from Camera
Image Resize
Histogram Equilisation
Data Store of RGB Image
Partition Training and Testing Data
Training CNN based Classifier
Do the Results match the desired Accuracy?
No
Yes
Fine tune the architecture and hyper-parameters of the proposed CNN
Classification using Trained CNN
Output

Fig. 1. Flowchart of proposed system

B. Methodology

Figure 1 depicts the working of the proposed system in flowchart form. The flowchart can be broken down into the following steps:

Step 1: The dataset is collected for training and validation.

Step 2: The collected dataset is pre-processed using Histogram Equalisation (HE) technique.

Step 3: Now the collected images will be converted into grayscale.

Step 4: These converted images will be converted to binary images. A threshold value is set so that the pixel value which is above a certain intensity are set to white and the rest which is below the value are set to black.

Step 5: The coordinates of the image are generated from binary images.

Step 6: The generated coordinates are then compared for output generation with the stored coordinates.

Step 7: The generated images are classified using CNN algorithm.

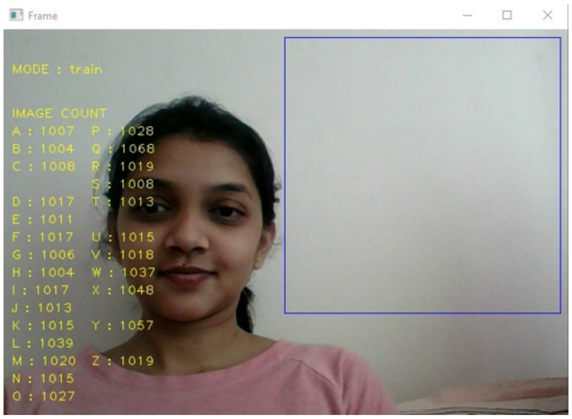
Step 8: The layers are built and real-time prediction is done.

Step 9: The predicted images are validated for real-time comparison.

C. Implementation

The implementation sequence can be broken down into the following steps:

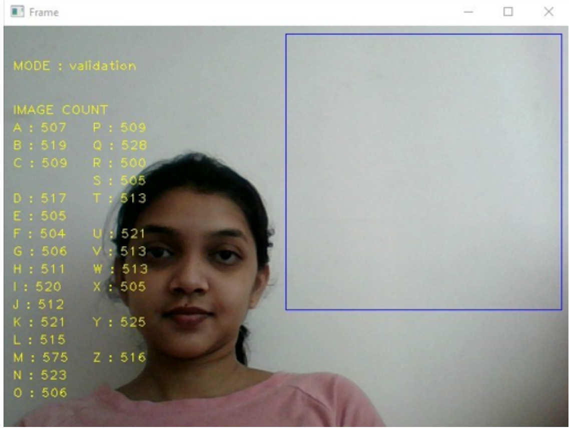
1) Tools: TensorFlow is a collection of processes for creating and training models in Python or JavaScript, as well as deploying them in the cloud, on-premises, in the browser, or on-device, independent of language. The purpose



```
MODE : train

IMAGE COUNT
A : 1007   P : 1028
B : 1004   Q : 1068
C : 1008   R : 1019
           S : 1008
D : 1017   T : 1013
E : 1011
F : 1017   U : 1015
G : 1006   V : 1018
H : 1004   W : 1037
I : 1017   X : 1048
J : 1013
K : 1015   Y : 1057
L : 1039
M : 1020   Z : 1019
N : 1015
O : 1027
```

Fig. 2. DataSet Collection for Train Set



```
MODE : validation

IMAGE COUNT
A : 507    P : 509
B : 519    Q : 528
C : 509    R : 500
           S : 505
D : 517    T : 513
E : 505
F : 504    U : 521
G : 506    V : 513
H : 511    W : 513
I : 520    X : 505
J : 512
K : 521    Y : 525
L : 515
M : 575    Z : 516
N : 523
O : 506
```

Fig. 3. DataSet Collection for Test Set

of Tensorflow is to construct a deep neural network that is capable of recognizing alphabets A-Z in sign language with reasonable accuracy. OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for image recognition and pre-processing of the dataset.

2) Dataset Collection: The ASL Alphabet, a dataset collection of American Sign Language (ASL), is where the original dataset is developed for this system. There is no standard dataset for ASL. As a result, original dataset has been created. The alphabets are among the gestures (A-Z). The images are grayscale and 320x240 pixels in size. The collection contains 39,948 photos with a resolution of 60x60 pixels. There are 26 classes in all, each comprising 26550 training images and 13398 test images. For greater precision, the data are recorded in several orientations. With photos acquired through laptop's webcam, this data is gestured in ASL. Cropped, rescaled, equalised, and labelled images are then used. Figures 2 and 3 depict the dataset collection for both train and test dataset.

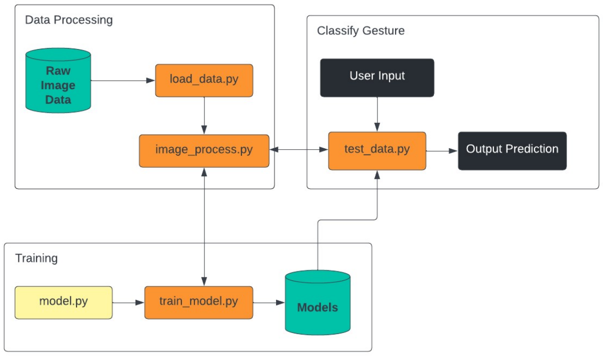


Fig. 4. Data Pre-processing

3) Data Pre-processing: The data pre processing technique enhances the relevant features as well as diminishing the irrelevant information or noise from the datasets. During data pre-processing, the unnecessary images are removed that are either misclassified. The foreground extraction technique is used so that only information related to the hand gesture is kept alone by removing the unnecessary background details. The foreground extraction technique assists in eliminating the background noise. The dataset collected are all of similar dimensions (64x64), however the external datasets used for evaluating the model all consisted of images with different resolutions thus requiring rescaling. Figure 4 depicts the data pre-processing of the collected dataset. The raw data images are loaded and preprocessed by image resizing and HE technique. The pre-processed images are used to train the model built with CNN layers.

D. Deep Learning Model

Once the dataset is ready, the proposed RTSLG system uses a combination of CNN architecture and Histogram Equalization(CNN+HE) as the deep learning model that learns patterns from the training data and uses these patterns to predict the alphabets. The performance of the network can be improved by hyperparameters tuning that reduces the problem of overfitting and underfitting. Figure 5 shows the architecture of the CNN model used in the proposed system. This model was made up of convolutional blocks that included two 2D convolutional layers with ReLU activation and softmax activation, as well as max pooling and dropout layers.

IV. Result Analysis

A. Real-Time Prediction

The model is tested in real-time and the results achieved are as follows. Figure 6 depicts the real-time predictions.

Figure 7 depicts the model accuracy. Model Accuracy is the measurement used to assess which model is the most accurate in identifying correlations and patterns between variables in a dataset based on the input, or training, data. The trained model has an accuracy of 87% for the best fit

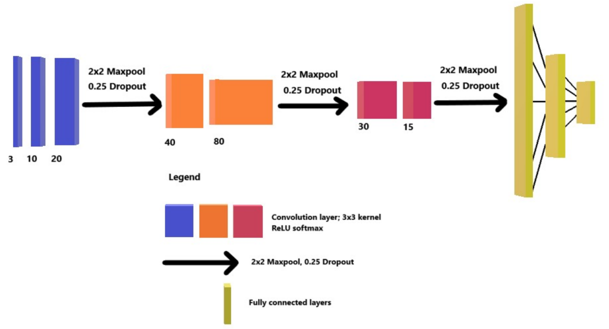


Fig. 5. Layers of CNN

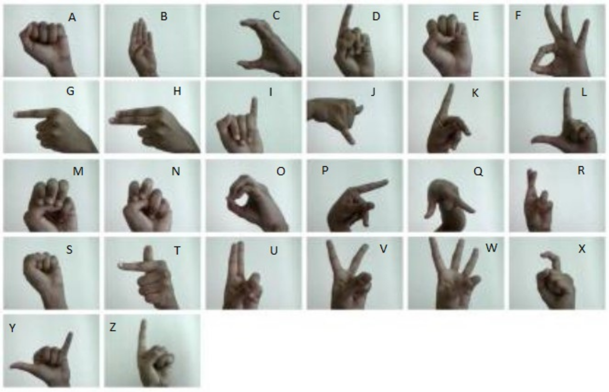


Fig. 6. Real-Time predictions

for the dataset, epoch 15. The accuracy can be improved however, it depends on the dataset and varied backgrounds. Figure 8 depicts the model loss. A loss is the penalty for making a wrong prediction. In other words, loss is a statistic that indicates how inaccurate the model's prediction was on a particular occasion. The loss is 0 if the model's prediction is accurate; otherwise, the loss is higher. X-axis corresponds to the number of epochs the model was trained on, while the Y-axis corresponds to the accuracy and loss of the model for the training and validation set respectively.

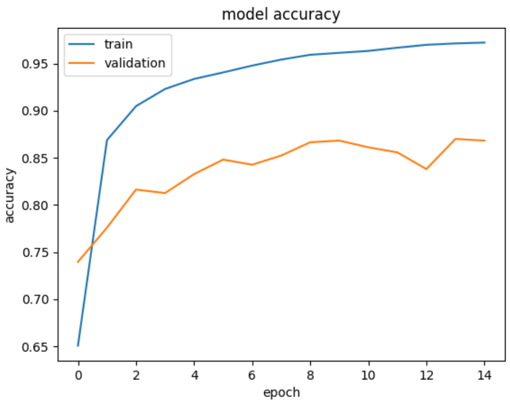


Fig. 7. Model Accuracy

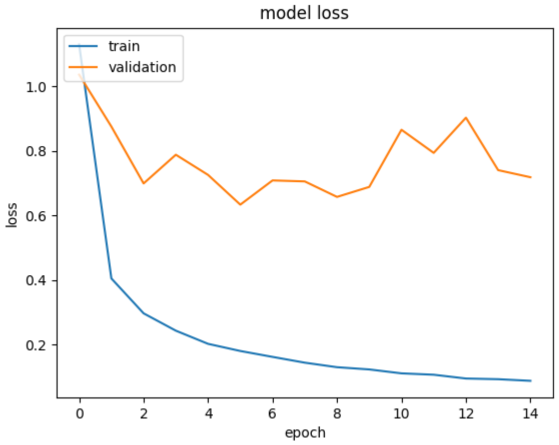


Fig. 8. Model Loss

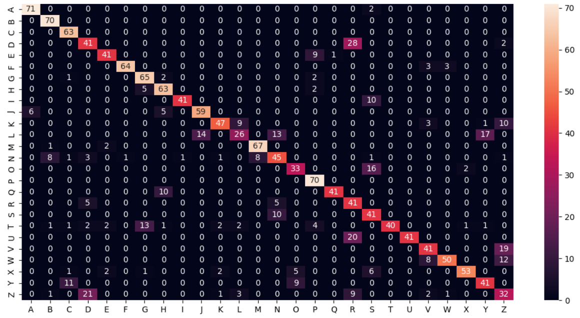


Fig. 9. Confusion Matrix

B. Validation

The dataset created is validated against an existing dataset. External Validation (EV) i.e: cross validation is the use of datasets obtained independently (thus, external) to validate the performance of a model trained on original input data. Here, for cross validation, an American Sign Language dataset is taken from Kaggle from the user Ayush Thakur [17] and over 80 test data are taken from each alphabets and the external validation is done on the same.

Figure 9 depicts the confusion matrix. The external testing dataset contains folders from A-Z, each folder contains 70 images that are checked against the dataset prepared. The outcomes are shown as a confusion matrix.

A confusion matrix summarizes the predicted outcomes for a classification issue. Each row of the matrix corresponds to an actual class, and each column corresponds to a forecast class. A diagonal across the matrix is beneficial since it indicates that classes were as expected. A confusion matrix is generated for CNN+HE, with test datasets for 26 classes and

TABLE I
ACCURACY COMPARISON FOR REAL-TIME SIGN LANGUAGE DETECTION

Number of epochs	CNN	Canny Edge Detection	CNN+HE (Proposed)
15	83%	60%	87%
20	84%	63%	87.0%
30	85%	63.5%	87.7%
40	85.5%	64%	87.8%

each class includes 70 dataset, the following classes showed anomalies: d, g, h, j, n, r, x i.e., these classes are showing some wrong predictions for some data. It can be observed that as the data for wrong predictions are less than 20%, the difference can be ignored.

C. Algorithm and Epoch Comparison

Table I shows the comparison between different algorithms such a CNN, Canny Edge Detection and the proposed algorithm CNN+HE. After implementing CNN, Canny Edge Detection and CNN+HE respectively, out of the three, the proposed algorithm CNN+HE performs the best because in the proposed system, the details of the images are enhanced, leading to achieve comparatively better accuracy. Histogram Equalization accomplishes this by effectively spreading out the most frequent intensity values, i.e. stretching out the intensity range of the image. Further for best fit model CNN+HE is used for building the model and training it in real-time for predictions.

D. Text-to-voice Output

Voice Output is produced using gTTS library and playsound, an command-line interface tool is used for interacting with Google Translates text-to-speech API and play the audio directly from the python program.

V. CONCLUSION AND FUTURE WORK

Intelligent systems in sign language recognition continue to attract the interest of academics and industry practitioners, thanks to recent advancements in machine learning and computational intelligence approaches. The proposed algorithm is highly efficient in predicting gestures from American sign language in real-time. These predicted alphabets are converted to form words and hence form sentences. The proposed system uses Histogram Equalization for image processing and then the processed images are used to train the model built using CNN layers. An original dataset is created for training and testing. For validation, an external dataset is used. Accuracy of CNN, Canny Edge Detection and the proposed CNN+HE is compared and the proposed system is found to have the highest accuracy out of the three of 87.8% for 40 epochs.

The proposed system can be developed and deployed utilizing Raspberry Pi in the future. The image processing element of the system should be upgraded so that it can communicate in both directions, i.e. it should be able to transform conventional language to sign language and vice versa. The signals that involve movement can be striven to be spotted. Furthermore, in future, the focus will be on turning the sequence of motions into text (words and phrases) and subsequently into audio.

REFERENCES

[1] A. Orbay and L. Akarun, "Neural sign language translation by learning tokenization," in *15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, 2020, pp. 222–228.

[2] W. Sandler and D. Lillo-Martin, *Sign language and linguistic universals*. Cambridge University Press, 2006.

[3] I. Farag and H. Brock, "Learning motion disfluencies for automatic sign language segmentation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 7360–7364.

[4] R. R. Koli and T. I. Bagban, "Human action recognition using deep neural networks," in *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, 2020, pp. 376–380.

[5] M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, M. A. Bencherif, T. S. Alrayes, H. Mathkour, and M. A. Mekhtiche, "Deep learning-based approach for sign language gesture recognition with efficient hand gesture representation," *IEEE Access*, vol. 8, pp. 192 527–192 542, 2020.

[6] D. Konstantinidis, K. Dimitropoulos, and P. Daras, "A deep learning approach for analyzing video and skeletal features in sign language recognition," in *2018 IEEE international conference on imaging systems and techniques (IST)*, 2018, pp. 1–6.

[7] Y. Yan, Z. Li, Q. Tao, C. Liu, and R. Zhang, "Research on dynamic sign language algorithm based on sign language trajectory and key frame extraction," in *2019 IEEE 2nd International Conference on Electronics Technology (ICET)*, 2019, pp. 509–514.

[8] H. Ma, Q. Wang, X. Ma, and M. E. Salem, "A sign language interaction system based on pneumatic soft hand," in *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2020, pp. 568–573.

[9] E. Abraham, A. Nayak, and A. Iqbal, "Real-time translation of indian sign language using lstm," in *2019 Global Conference for Advancement in Technology (GCAT)*, 2019, pp. 1–5.

[10] R. Rastgoo, K. Kiani, and S. Escalera, "Hand sign language recognition using multi-view hand skeleton," *Expert Systems with Applications*, vol. 150, p. 113336, 2020.

[11] S. S. Rautaray and A. Agrawal, "Interaction with virtual game through hand gesture recognition," in *2011 International Conference on Multimedia, Signal Processing and Communication Technologies*, 2011, pp. 244–247.

[12] N. M. Adaloglou, T. Chatzis, I. Papastratis, A. Stergioulas, G. T. Papadopoulos, V. Zacharopoulou, G. Xydopoulos, K. Antzakas, D. Papazachariou, and P. none Daras, "A comprehensive study on deep learning-based methods for sign language recognition," *IEEE Transactions on Multimedia*, 2021.

[13] K. Zhao, K. Zhang, Y. Zhai, D. Wang, and J. Su, "Real-time sign language recognition based on video stream," *International Journal of Systems, Control and Communications*, vol. 12, no. 2, pp. 158–174, 2021.

[14] Y. Chen and W. Zhang, "Research and implementation of sign language recognition method based on kinect," in *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, 2016, pp. 1947–1951.

[15] S. A. Khan, A. D. Joy, S. Asaduzzaman, and M. Hossain, "An efficient sign language translator device using convolutional neural network and customized roi segmentation," in *2019 2nd International Conference on Communication Engineering and Technology (ICCET)*, 2019, pp. 152–156.

[16] C. Chansri, J. Srinonchat, E. G. Lim, and K. L. Man, "Low cost hand gesture control in complex environment using raspberry pi," in *2019 International SoC Design Conference (ISOCC)*, 2019, pp. 186–187.

[17] "Asl external dataset," https://www.kaggle.com/ayuraj/asl-dataset.