Engineering and Scientific International Journal (ESIJ)
Volume 9, Issue 1, January – March 2022
ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

Deep Learning based Speech and Gesture Recognition System for the Disabled

Sheena Christabel Pravin[1*], Saranya.J[1], M. Palanivelan[2], Priya L[3]
[1]*Assistant Professor, Department of Electronics and Communication Engineering, Rajalakshmi Engineering College,*
[2]*Professor, Department of Electronics and Communication Engineering, Rajalakshmi Engineering College,*
[3]*Professor, Department of Information Technology, Rajalakshmi Engineering College, Chennai*
Corresponding Author Email-id: sheena.s@rajalakshmi.edu.in

Abstract— Speech and Gesture recognition systems constitute an ideal aid for the disabled with speech and hearing impairments. Approximately, there are 466 million people in the world with hearing impairment and around 16 million with speech impairment. They require an external aid to recognize their speech and gestures, to express their thoughts and ideas to the world. The proposed Speech and Gesture Recognition System (SGRS) takes forward to solve the communication barriers faced by the disabled subjects, by recognizing both the speech and gestures of the subjects with promising accuracy using the convolutional neural network. The proposed SGRS model is competent to convert the sign-language into pictures and speech to text as well with high accuracy. Thus, SGRS can be a suitable aid for the subjects with speech and hearing impairment. SGRS has been evaluated with standard evaluation scores such as validation accuracy, validation loss, recall, precision and F1-score and has been proved to be proficient.

Keywords *— House Speech and Gesture Recognition System; Deep Learning; Convolutional Neural Network; Speech and Hearing Impairment.*

1. Introduction

Sign language is a customized communication aid, that utilized symbolic approach like hand gestures to express meaning. Sign language has proved to be tremendously supportive for subjects with difficulties in speaking or hearing. Very few are trained to interpret sign language. Thus, machine learning based gesture recognition systems are gaining popularity to aid the disabled in seamless communication. Sign language recognition denotes to the translation of hand gestures into text of a syntactically spoken languages. Thus, translation of sign language into text by a machine learning model can aid to tie the people with hearing or speaking disabilities to the rest of the world.

Numerous techniques have been experimented to translate sign language to text in the yester years. Conventional Sign language translation techniques consists of two steps, namely there cognition of each hand gesture in the given image, followed by classification of the same into the relevant text. Leap motion-based hand tracking system [16] use machine learning algorithms like Support Vector Machines to classify hand gestures. Kinetic sensors are hardware devices that are serve as 3D models of the hand and deciphers the hand movements with their diverse orientations.

A glove-based technique insists that the user wears a special glove that detects the position and diverse orientations of the hand. Hardware systems have been found to be accurate, but they incur initial setup cost and are expensive. A few previous methods used computer vision procedures viz. the convex hull method to regulate the convexness in the given image and perceive the edges of the hand in the given image. There are also silhouette based procedures which look for skin-like silhouettes in the image to sense the hand. The recognition is later tailed by machine learning models that have been trained to classifying the image of the hand gestures into text. In [1], the Convolutional Neural Network, when trained on the American Sign Language dataset attained high test accuracy. Yet, gesture recognition was not parallelized with speech recognition to aid the hearing impaired. Basic Component Analysis based feature vector extraction [2] and Hidden Markov Model based classification brought fruitful sign language recognition while k-Nearest Neighbors classifier [3] was also experimented in recognizing the gestures presented in the American sign-language dataset. Edge detection and cross-correlation techniques were employed in [4], to interpret gestures. Inception v3 model on a custom dataset [5], yielded a validation accuracy of 90% in recognition of sign language.

The structure of the article is as follows: Section II illustrates the dataset wrought for this research work. Section III delineates the process flow of the new Speech and Gesture Recognition System, proposed in this work. Section IV presents the evaluation results of the proposed model with Conclusions in Section V.

2. Dataset

The dataset was fashioned by hand due to the lack of a whole dataset for Sign Language which included all the 26 letters of the English Language. The dataset contains

DOI: **10.30726/esij/v9.i1.2022.91002**

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

almost 5000 images for each letter in diverse circumstances and under different lighting environments. The images for individual letters were augmented to enhance the size and efficiency of the dataset. The images were improvised, rotated, and shifted to introduce images for each letter which were diverse with respect to the context, orientation, and lighting circumstances.

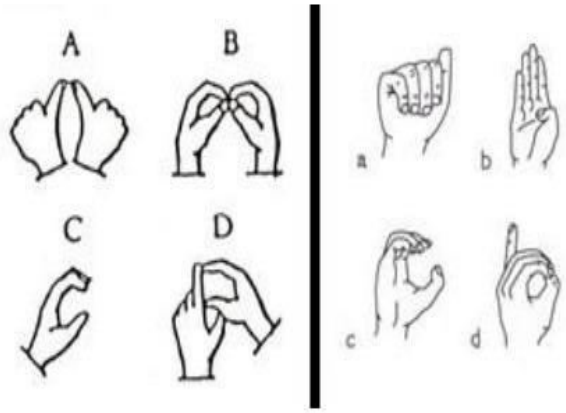

Fig. 1: (a) American Sign Language (b) Indian Sign Language Sample Gestures [5]

The illustration of American Sign Language (ASL) is shown in Figure 1(a), while Figure 1(b) described the Indian Sign Language (ISL) datasets. The ASL and the ISL Datasets had less data samples and had few two-handed gestures and dynamic gestures, we chose to create our own dataset. The dataset created by this research work has 27(alphabets + space) classes with 5000 images per class.

Fig.2: Custom Dataset

Custom dataset derived from the actual dataset is shown in Figure 2. The images are then pre-processed by using Canny Edge Detection to make the gestures easily distinguishable and thereby increasing the accuracy of the model. Images are canny edge detection process is shown in Figure 3.

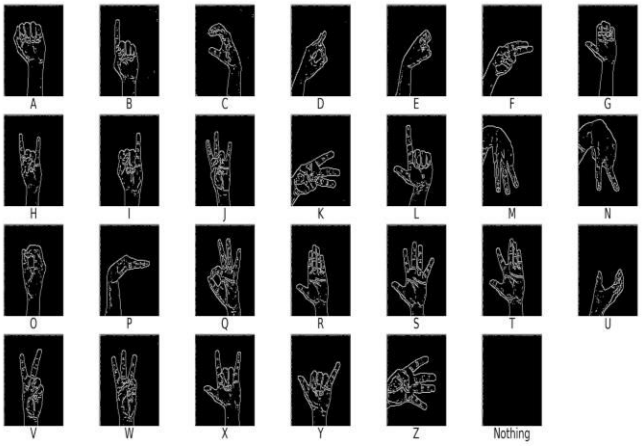

Fig.3: Custom Dataset after Initial processing (Canny)

3. Methodology

Computer Vision is perhaps the most intriguing and fascinating concept in artificial intelligence. Computer Vision deals with higher dimensional understanding of images and their features by machine learning and deep learning models by visualization to solve problems pertaining to image classification, image enhancement and resynthesis. After obtaining this conceptual perspective, it can be useful to automate tasks or perform the desired action [6].

The obvious functions in the human brain are less sensitive to computers as they require specialized training in these functions in order to produce effective results. This process involves complicated steps like acquiring the data from the real world, processing the acquired data in a suitable format, analyzing the processed images, and finally teaching and training the model to perform the complex task with very high accuracy [6]. The OpenCV module is the best module for complex computer learning, in-depth learning, and computer programming activities. Provides simplicity and high levels of analysis and performance of built-in models. It is an open library and can be integrated with other python modules like NumPy to achieve complex real-time applications. It is supported for a wide range of programming languages and runs remarkably on most platforms such as Windows, Linux, and MacOS.[6]

4. Speech & Hand Gesture Recognition System

The overall system framework for the process of speech recognition is shown in Figure 4. The speech signals are first pre-processed using suitable pre-emphasis filter, followed by speech segmentation and framing for feature extraction. The speech feature patterns are

DOI: 10.30726/esij/v9.i1.2022.91002

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

compared with the pre-defined phonetic units. After relevant word and sentence match, the speech to text converter produces the recognized text. In real time, the test speech is rapidly converted to text using the speech to text converter.

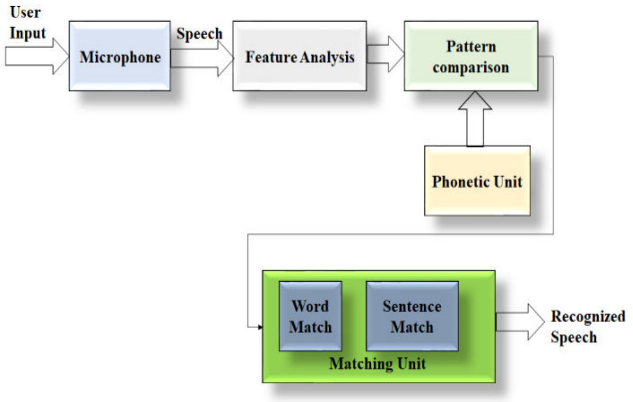

Fig.4: Speech recognizer

The process of gesture to text is shown in Figure 5. The hand image is captured by a high-resolution camera. The region of interest is apprehended by segmentation. The process of normalization assures the centering of features around the mean and variance nearing 1. The convolutional neural network, a deep learning network does feature engineering by itself and extracts high-dimensional features for gesture image classification to relevant text output. The text to speech converter aids in converting the text output of the CNN to speech.

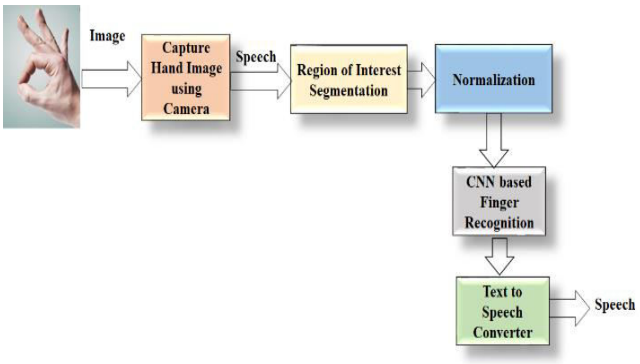

Fig.5: Deep learning-based Hand Gesture to Speech System

4.1 Convolutional Neural Network

Convolutional Neural Networkis a set of deep neural networks which is mostly handed down to do image recognition, image classification, object detection, etc.[7]. The progression in Computer Vision has been built and achieved with deep learning models such as the convolutional neural network [7].

Image classification is the machine learning task of training the model to bring out the probability of the defined classes which best fits the image. In CNN, an image is given as input, append weightage to its diverse features to differentiate one image from the other for classification. The preprocessing required in CNN is much lesser as compared against other classification algorithms as it is a deep learning model and it strives to compute features from the input image [7].Computers cannot see things as we do, for computers image is nothing but a matrix. A CNN characteristically has 3 layers, namely the convolutional layer, max/min pooling layer(s), and a final fully connected softmax layer for classification [7].

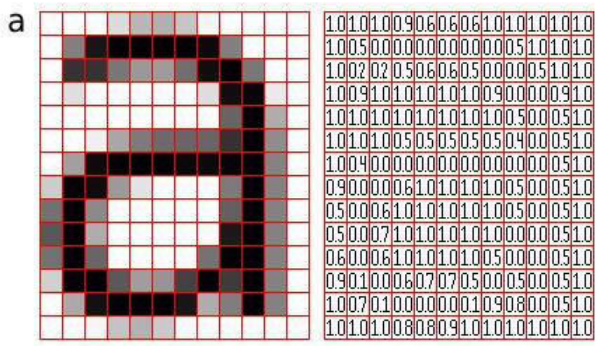

Fig.6: Matrix representation of letter a [7]

Sample output for the letter a through CNN model is shown in Figure 6. The convolution layer is the central structural module of CNN. It transfers the significant portion of the network's computational load. The essential objective of convolution is to retrieve features such as edges, colors, corners from the input. As we go deeper inside the network, the network starts identifying more complex features such as shapes, digits, face parts as well. [7]

Pooling layer decreases the computational power obligatory to process the data. It is done by decreasing the dimensions of the featured matrix even more. In this layer, we try to extract the dominant features from a restricted amount of neighborhood [7]. Flattening is essential to convert the data into a 1-dimensional array for sending it as an input to the adjacent layer. In this work, the output of each convolutional layers is flattened to build a unitary feature vector. It is then connected to the fully connected classification layer of the prediction model [8]. Fully Connected layer is simply a feed forward network. The last few layers in the network form the fully connected layers. The final Pooling layer gives the input to the fully connected layer for generating the final prediction probabilities to determine the class label [9].

A lot of theory and mathematical machines behind the traditional machine learning models like the support vector machines, were developed with linear models in mind.

DOI: 10.30726/esij/v9.i1.2022.91002

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

However, practical real-life problems are often nonlinear in nature and, therefore, they cannot be effectively solved. The non-linear activation function like ReLU, Softmax, Leaky-ReLU, Sigmoid, Tanh, etc. [10] is applied over the output data from a particular layer of neurons before it propagates as the input to the next layer. Optimizers Adam, Adagrad, Nadam, etc. [11] are algorithmic methods that are used to modify the parameters of the neural network viz. weights, biases and learning rate to decrease the losses.

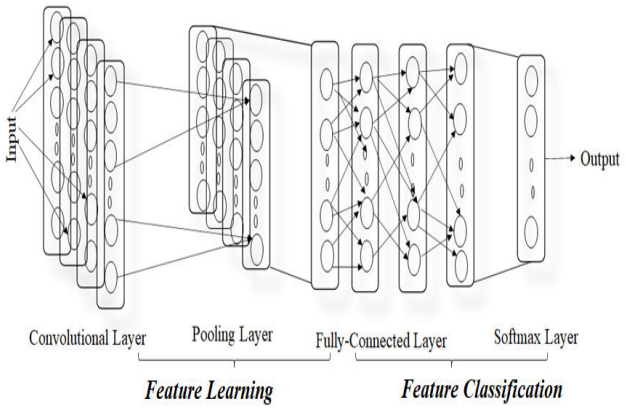

Fig.7: CNN Layers for feature learning and classification

The proposed model has four Convolutional Layers and two fully connected layers, each with ReLU Activation layer, batch normalization layer, max-pooling layer and dropout layer. Various layers of CNN model are presented in Figure 7. The convolution layer output is flattened before sending into the fully connected layer. The output layer has 27 possible outcomes and has a Softmax Activation Layer. The loss monitored is categorical cross entropy, while the Optimizer used is Adam. The guarded loss is a cross entropy phase, while the Optimizer used is Adam. ReLU indirect opening function that began to thunder in the context of the Convolution Neural Network. If the input is positive then the function would output the value itself, if the input is negative the output would be zero. [10]

The softmax function is used as the activation function at the output layer of neural network models to predict a multinomial probability distribution [18]. That is, softmax is used as the activation function for multi-class classification problems [12]. Adam is anauxiliary optimization algorithm to stochastic gradient descent to train the deep learning models. Adam agglomerates the unique features of AdaGrad and RMSProp algorithms to handle scant gradients on noisy problems.[12]

Batch normalization allows every layer of the network to learn independently. It is used to normalize the output of the previous layers. The activations scale the input layer in normalization. Using batch normalization learning becomes efficient. Also, it serves as a regularization unit to prevent overfitting of the model [13].

Dropout is a form of regularization and is highly recommended to avoid overfitting of the model. Dropouts are appended arbitrarily by nullifying some percentage of neurons of the network. When the neurons are nullifying, the incoming and outgoing connections to those neurons are also nullified. This enhances the learning of the model [13].

Categorical cross entropy is a loss function used in multiple class division operations. These are functions where the model can only be one of the many possible types, and the model must determine which one. Officially, it is designed to measure the difference between the two distributions of opportunities.

4.2 Speech - Text Conversions

The CNN model gives us an output in text format which is then converted into audio. This conversion is done using the python library pyttsx3. Unlike alternative libraries, it works offline and is compatible with both Python 2 and 3. The audio input is converted into text by using the python library Speech Recognition. It can support various API's like Google Cloud Speech API , Wit. Ai , etc.. The python library PyAudio is required if and only if you want to use microphone input. Flow of process for both speech to gestures and Gestures to speech is presented in Figure 8 and Figure 9.

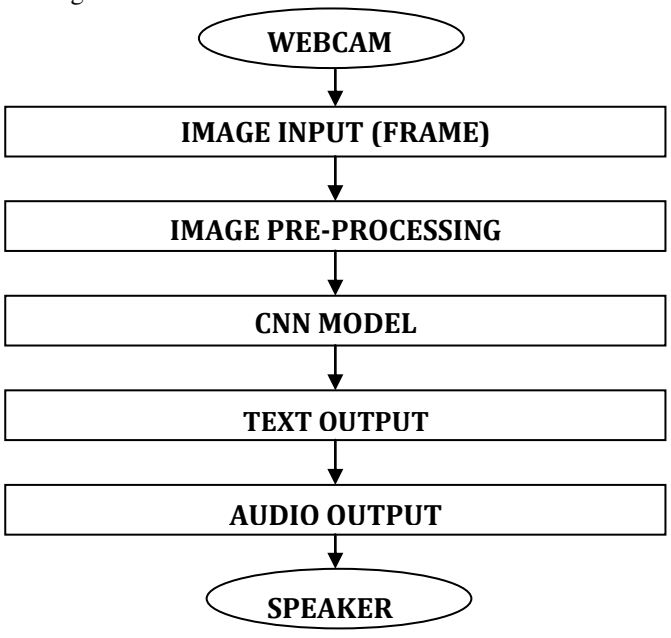

Fig. 8: Gestures to Speech

Pseudo code for speech to gesture is given in Figure 9 for better understanding of the proposed model.

DOI: 10.30726/esij/v9.i1.2022.91002

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

Pseudocode for Speech and Gesture Recognition

```
import necessary packages
define class
    create objects for XML design
    set necessary permissions
store images and word or sentences in backend
allow the User to provide input as speech
convert speech to string
if string matches with stored alphabets
    then:
    display appropriate image
end
```

Fig. 9: Pseudocode for proposed SGRS model

After getting audio input from the user, the system converts the input into text and split the text into alphabets, these alphabets are then matched with the existing stored alphabets, then the corresponding image will be displayed. The same is depicted as flow diagram in Figure10.

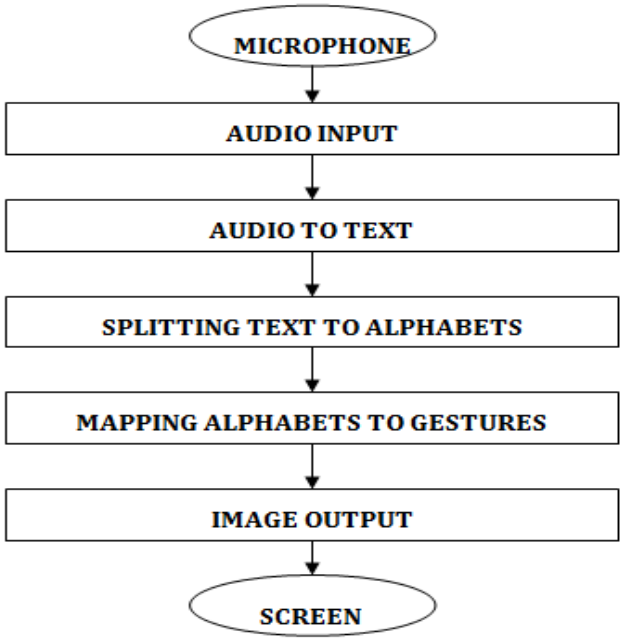

Fig. 10: Speechto Gestures

4.3 Graphic User Interface

Kivy is a Python image source for user open source software that allows you to develop multi-platform applications on Windows, macOS, Android, iOS, Linux, and Raspberry-Pi. In addition to standard mouse and keyboard input, it also supports multi touch events. Apps made using Kivy will be the same in all forums but it also means that the apps feel or look will be different from any traditional application. A sample GUI is shown below in Figure 11.

Fig. 11: App GUI

5. Results

The process of speech to gesture and gesture to speech is shown in Figures12and 13. From the figures, it is understood that, the proposed system is assisting at a higher degree to the physically challenged people. This in turn enables easy communication for the physically deprived.

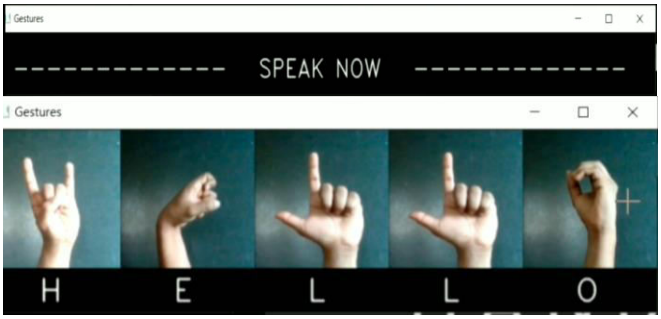

Fig. 12: Speech to Gestures

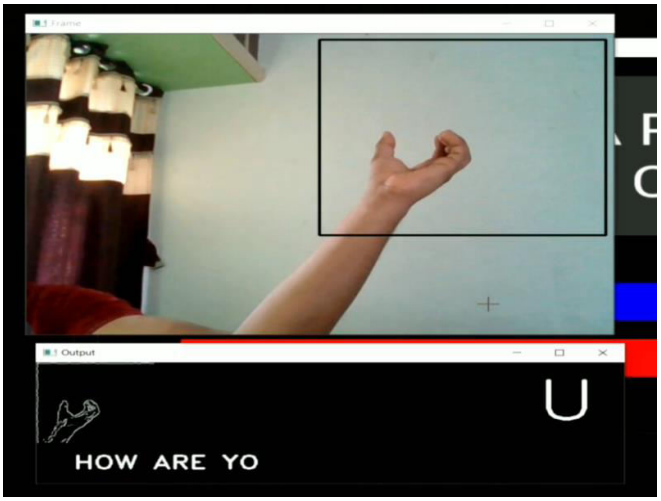

Fig. 13: Gestures to Speech

DOI: 10.30726/esij/v9.i1.2022.91002

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

The proposed model uses CNN as mentioned in the previous sessions. The CNN model was trained for using multiple optimizers namely Adam, Nadam and RMSprop. Below figures show the validation accuracies and validation losses for the different optimizers used.

Model performance measures such as Validation accuracy, Validation losses with respect to Epochs, are calculated and shown in Figures (14) to (17). From the figure it has been identified that, the model provides better performance measure rates for recall, precision and F1 scores, which reveal the proficiency of the model.

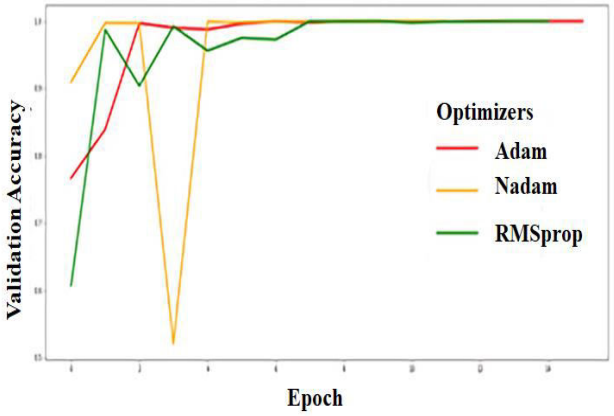

Fig. 14: Validation Accuracies for Different Optimizers

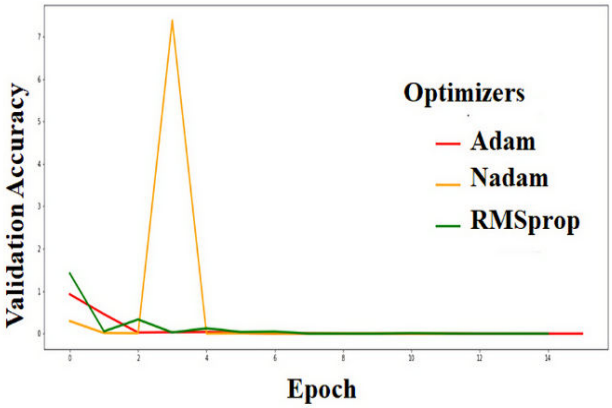

Fig. 15: Validation Losses for Different Optimizers

From the figures, it is understood that, all the three optimizers yielded 100% test validation accuracies. RMSprop took longer convergence time to give us the best results and Nadam had the problem of overfitting initially before giving us the best result. Therefore, the Adam optimizer was chosen, since it had the least convergence time and suffered less overfitting.

The training accuracy, validation accuracy of the model, along with the loss function using Adam optimizer is shown below in figures 16 and 17 respectively. Best validation accuracy and loss for Adam optimizer is obtained around the 6th epoch as shown in Figure 16. Also, the loss function, when plotted with respect to Epochs shows a huge dip in training and validation loss as the number of epochs increases. This proves the generalization ability of the model.

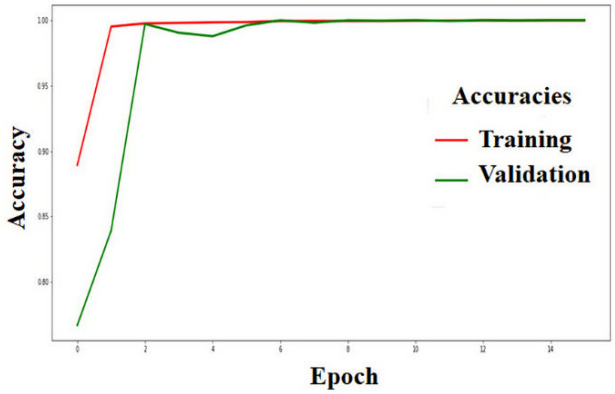

Fig. 16: Accuracy v/s Epochs for Adam

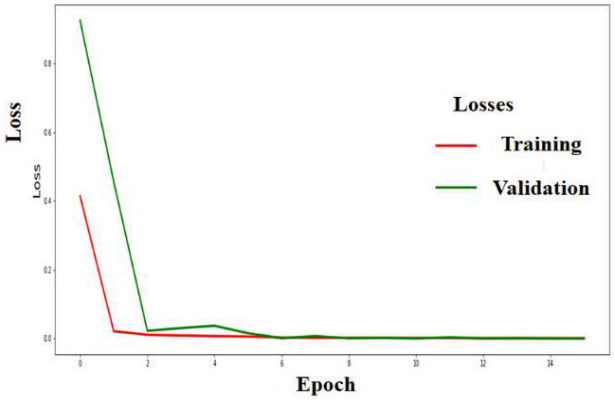

Fig. 17: Loss v/s Epochs for Adam

The proposed model has been evaluated with respect to the additional evaluation metrics viz, recall, precision and F1-score. Precision is used to quantify the true positive class labels that are correctly predicted from the total predicted outcomes. It is the ratio of the rightly categorized positive observations to the total positive reflections. Precision was measured as given in equation (1)

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (1)$$

High precision corresponds to low false positive estimate. Thus, precision is a performance metric of the classifiers which describes the reliability of the model in identifying the true positives right.

Recall is an estimate of the fraction of positive class labels that are rightly predicted. It is the ratio of accurately projected positive class samples to all the actual positive labels. It was measured as in equation (2)

DOI: 10.30726/esij/v9.i1.2022.91002

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (2)$$

Recall score describes the sensitivity of the model to true positives and is a direct measure of true positive data samples that are rightly classified by the model. F1 score is the harmonized average of recall and precision scores [17]. Also, the F1 score was computed as given in equation (3).

$$\text{F1 score} = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (3)$$

The complete classification account of the proposed model is given in Figure 18, which displays the higher proficiency of the model.

CLASSIFICATION REPORT : MODEL 6

	precision	recall	f1-score	support
A	1.00	1.00	1.00	1000
B	1.00	1.00	1.00	1000
C	1.00	1.00	1.00	1000
D	1.00	1.00	1.00	1000
E	1.00	1.00	1.00	1000
F	1.00	1.00	1.00	1000
G	1.00	1.00	1.00	1000
H	1.00	1.00	1.00	1000
I	1.00	1.00	1.00	1000
J	1.00	1.00	1.00	1000
K	1.00	1.00	1.00	1000
L	1.00	1.00	1.00	1000
M	1.00	1.00	1.00	1000
N	1.00	1.00	1.00	1000
Nothing	1.00	1.00	1.00	1000
O	1.00	1.00	1.00	1000
P	1.00	1.00	1.00	1000
Q	1.00	1.00	1.00	1000
R	1.00	1.00	1.00	1000
S	1.00	1.00	1.00	1000
T	1.00	1.00	1.00	1000
U	1.00	1.00	1.00	1000
V	1.00	1.00	1.00	1000
W	1.00	1.00	1.00	1000
X	1.00	1.00	1.00	1000
Y	1.00	1.00	1.00	1000
Z	1.00	1.00	1.00	1000
accuracy			1.00	27000
macro avg	1.00	1.00	1.00	27000
weighted avg	1.00	1.00	1.00	27000

Fig. 18: Classification Report

6. Conclusion

This work proposed a interactive Speech and Gesture Recognition system. It has been trained and tested on a self-created sign language dataset using OpenCV library in Python. The proposed Speech and Gesture Recognition Model for sign language classification was successfully built as a communication aid for the speech and hearing impaired using the convolutional neural network. The proposed model showed a validation accuracy, which was nearly 100% with a real time accuracy of 93%. The model can further be enhanced to read the sentences of 1000 words with a higher classification model using text mining.

Conflict of Interest

The authors declare no conflict of interest.

Funding Source

This work was partially funded byAICTE, India, under the Research Promotion Scheme (The Grant Reference No. is: 8-40/RIFD/RPS/Policy-1/2017-18, dated 15 March 2019).

Reference

[1] TaskiranM, KilliogluM and Kahraman N. A Real-Time System for Recognition of American Sign Language by using Deep Learning, 2018; 41st International Conference on Telecommunications and Signal Processing (TSP), Athens, 2018, pp. 1-5.

[2] Zaki,M M, Shaheen S.I. Sign language recognition using a combination of new vision based features; Pattern Recognition Letters, 2011;201132 : 572–577.

[3] Aryanie D, Heryadi Y. American sign language-based finger-spelling recognition using k-Nearest Neighbors classifier, in Proc. 3rd International Conference on Information and Communication Technology.

[4] Joshi A, Sierra H, Arzuaga, E. American sign language translationusing edge detection and cross correlation, in Proc. IEEE Colombian Conference on Communications and Computing (COLCOM), Cartagena,Colombia, 2017; 1–6.

[5] Das A, Gawde S, Suratwala K and Kalbande D. Sign Language Recognition Using Deep Learning on Custom Processed Static Gesture Images, 2018 International Conference on Smart City and Emerging Technology (ICSCET), Mumbai, 2018; 1-6.

[6] Barath K, OpenCV: Complete Beginners Guide To Master Basics Of Computer Vision With Codes, 2020; Available: https://towardsdatascience.com/opencv-complete-beginners-guide-to-master-the-basics-of-computer-vision-with-code-4a1cd0c687f9.

[7] Manan Parekh. A Brief Guide to Convolutional Neural Network(CNN), 2019; Available: https://medium.com/nybles/a-brief-guide-to-convolutional-neural-network-cnn-642f47e88ed4

[8] Jiwon Jeong. The Most Intuitive and Easiest Guide for Convolutional Neural Network, 2019; Available: https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480#:~:text=Flattening%20is%20converting%20the%20data,called%20a%20fully%2Dconnected%20layer.

[9] Arunava. Convolutional Neural Network, 2019; Available: https://towardsdatascience.com/convolutional-neural-network-17fb77e76c05#:~:text=Fully%20Connected%20Layer%20is%20simply,into%20the%20fully%20connected%20layer.

[10] Kevin Vu. Activation Functions and Optimizers for Deep Learning Models, 2019; Available: https://dzone.com/articles/activation-functions-and-optimizers-for-deep-learn#:~:text ReLU%20is%20a%20non%2Dlinear,the%20output%20would%20be%20zero.

[11] Nagesh Singh Chauhan. Optimization Algorithms in Neural Networks, 2020; Available: https://www.kdnuggets.com/2020/12/optimization-algorithms-neural-networks.html#:~:text=Optimizers%20are%20algorithms%20or%20methods,problems%20by%20minimizing%20the%20function.

[12] Jason Brownlee. Softmax Activation Function with Python, 2020; Available: https://machinelearningmastery.com/softmax-activation-function-python/#:~:text=The%20softmax%20function with%20is%20used%20as%20the%20activation%20function%20in,more%20than%20two%20class%20labels.

[13] Rohit Dwivedi. Everything You Should Know About Dropouts and BatchNormalization In CNN, 2020; Available: https://analyticsindiamag.com/everything-you-should-know-about-dropouts-and-batchnormalization-in-cnn/

[14] Evgeny A. S, Denis M. Serge N.A. Comparison of Regularization Methods for ImageNet Classification with Deep Convolutional Neural Networks, 2014; 6:89-94.

DOI: 10.30726/esij/v9.i1.2022.91002

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

[15] Anup Kumar, Karun T.M, Dominic M, Sign Language Recognition, in Recent Advances in Information Technology, 3rd International Conference, 2016.
[16] Vysocký A, Grushko S, Oščádal P, Kot T, Babjak J, Jánoš R, Sukop M, Bobovský Z. Analysis of Precision and Stability of Hand Tracking with Leap Motion Sensor. Sensors. 2020; 20(15):4088.
[17] Sheena Christabel Pravin, Palanivelan, M. Regularized Deep LSTM Autoencoder for Phonological Deviation Assessment. International Journal of Pattern Recognition and Artificial Intelligence, 2021; 35(4): 2152002.
[18] Sheena Christabel Pravin, Palanivelan, M. A Hybrid Deep Ensemble for Speech Disfluency Classification. Circuits, Systems, and Signal Processing, Springer, 2021; 40 (8): 3968-3995

DOI: **10.30726/esij/v9.i1.2022.91002**