2022 IEEE World Conference on Applied Intelligence and Computing (AIC) | 978-1-6654-7988-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/AIC55036.2022.9848844

Sign Language Detection using Convolutional Neural Network (CNN)

Nipun Jindal, Nilesh Yadav, Nishant Nirvan, Dinesh Kumar
Electronics and Communication Engineering
Delhi Technological University
New Delhi, India
nipunjindal_2k18ec112@dtu.ac.in, nileshyadav_2k18ec111@dtu.ac.in, nishantnirvan_2k18ec115@dtu.ac.in, dineshkumar@dtu.ac.in

***Abstract* — Communication is important to express feelings of oneself. Effective communication helps in personal and professional growth. Communicating with a person having some disabilities, such as speech and hearing impairment, is always a major challenge. Deaf And Dumb people cannot communicate with person with no disability because of communication barrier between them and the other person not knowing the sign language. As Per India's Census 2011, At all India levels, disabled people constitute 2.21% of the total population. In India, about 19% of disabled people have hearing disabilities and 7% in Speech impairment [1]. Sign language gestures are not always enough for communication of people with hearing disability or people with impairment of speech. The gestures/signs made by the people having disabilities often get mixed or difficult to understand for someone who does not understand the language. Thus, we have implemented two models to convert sign gestures to text using Convolutional Neural Network (CNN) in Python and AlexNet in MATLAB.**

Keywords—Hand gesture, Sign language, ISL (Indian Sign Language), Manual features (MF) AlexNet, CNN, MATLAB.

I. INTRODUCTION

Sign language is a way to communicate using hand gestures which helps people with hearing/speech impairment to communicate. Deaf and Dumb people make up to 9% of the Indian population and 6% worldwide population. A huge segment of the world population is facing problems in order to understand and interpret what deaf and dumb people want to convey to the whole world. People with disabilities use different sign languages across different nations like American Sign Language (ASL), Indian Sign Language (ISL) and British Sign language (BSL). Although this mismatch of sign language creates confusion in interpreting across countries.

Thus, we have introduced a unified method of converting sign language into text where a person can create different gestures regardless of the sign language and can convert his thoughts into text for better communication.

We have presented two models, the first of which uses MATLAB to construct a dataset of 7 distinct motions and then uses AlexNet to recognize them. The motive behind using AlexNet was to make the model train faster as Alexnet divides training dataset onto the system GPU sections and helps to provide decent results with image recognition. Focus was to develop a system with good results and accessible at smaller devices such as raspberry pi and Arduino. This model was able to detect gestures with 70 percent accuracy, but it was more susceptible to noise while doing so. Later, utilizing Convolutional Neural Networks, a deep learning project was presented to recognize 20 separate motions with a single hand to avoid overlapping movements (CNN). In terms of performance, the model was on par, with a training accuracy of 98 percent and a testing accuracy of 94 percent. In both the models, the image is captured in real time through the webcam and then preprocessed such as thresholding, segmentation, noise removal and edge detection. In the testing stage, the detected gesture is shown on the live feed image with predictions on each frame that the live cam captures.

II. LITERATURE REVIEW

Various techniques have been deployed for implementation of sign language recognition. This paper thus touches upon these techniques and algorithms in order to understand these methodologies and find their drawbacks.

Thus, by providing a better solution using MATLAB and CNN to get the best results possible. Mahesh Kumar NB et al [1] have performed Indian Sign Language (ISL) recognition by computing d-dimensional mean vectors, scatter matrices and using eigenvalues and eigenvectors in the Linear Discriminant Analysis (LDA) algorithm which results in better dimension reduction. Kumud Tripathi et al. [3], in Principal Component Analysis (PCA), the other distance classifiers will be recognizing the hand gesture. Here the features of the keyframes will be extracted from the data set and are fed as input from the orientation histogram. Noor Tubaiz et al. [4] uses The Glove based approach which simplifies the task during segmentation process, here Modified k-Nearest Neighbor (MKNN) will be used for classification of the dataset. gloves read the hand motions this data was further improved with the calculated vectors features. B. Bauer et al. [6], employed continuous hidden markov models' images (HMM) for continuous detection of sign language and recognition, with input being those vectors that are used for showing manual signs. Wysoski et al. [7]

978-1-6654-7988-2/22/$31.00 ©2022 IEEE
DOI: 10.1109/aic.2022.59

used standard contour tracking algorithm, rotational invariant postures. The given picture was applied with an identification for skin color filter and setting up of grids which was then grouped to identify the border within every pooling image using the contour algorithm. Method employed by Hasan [9] was to determine the center point of a hand by shifting the location of any selected image to overlap the center point of hand so that the gestures are recognized by brightness factor matching. Ashish et al. [10] proposed an image detection model by using a cam and the preprocessing of the gestures is performed by using VS code (IDE) and OpenCV python library. The model generates a template of the captured image based on complex hull algorithm generated ratios. Skin segmentation and Color filtering are performed by converting RGB based images to HSV based images. Sharmila et al. [11] proposes an HSV color model that detects the color of human skin from the given image. Edge detection is applied to detect the shape of the hand by analyzing the edges from the image and output is generated after some morphological operations on the input image. The images were captured in variant environments and different geometric conditions to train the model with any edge case. The system shows an overall success rate of 65% under testing. Sajeena et al. [12] proposes a MATLAB based hand gesture recognition technique to convert sign language into text using AlexNet. The image is captured through the webcam and goes through a series of preprocessing such as noise removal, canny edge detection and shadow removal after conversion of RGB to HSI model. The model shows an overall accuracy of 75% under testing.

III. DATASETS

Sign language is based on hand gestures, and it can be challenging to find a proper dataset as manual features (MF) such as body posture, hand gesture and hand posture could vary from person to person. Thus, in order to provide a better solution, we have created our own dataset using digital camera/WebCam.The first proposed model using AlexNet consists of a total of 2100 images; 300 images for 7 distinct gestures.

For both the models a customized dataset of 2100 and 24000 images each has been created. The dataset was then preprocessed for better feature extraction as mentioned in fig. 4.

The second model using CNN and openCV uses a dataset of 24000 images: 1200 images for 20 distinct gestures. The dataset is divided in 8:2 proportion for training and validation. Thus, providing 19200 images for testing and 4800 for calculating the validation accuracy. For better and reliable results, we have also created a separate dataset for testing the machine that consists of 6500 images in totality for all the gestures i.e from 0-19 in random orientation and sizes as well.

IV. METHODOLOGY

Using AlexNet in MATLAB

A. Image Acquisition

In the acquisition stage, the images are captured with a webcam. The gestures are captured with bare hands in a fixed stable background rather than using glove-based technique. For the database creation as many as 300 image frames are captured from the live video feed.

B. Preprocessing

The images captured through the webcam are then resized to further preprocess them in order to train the dataset. The preprocessing includes noise removal using a low pass filter, removal of edge detection and shadow removal. All the images are converted to greyscale, thus reducing its size for fast processing The RGB images are resized and then transformed into HSI based model for well-defined image segmentation. The range of pixels from the HSI cone are calculated to determine the angle and skin color of the hand.

C. Segmentation

Segmentation refers to the segregation of images captured into the set of different parts by combining pixels to form a group on homogeneity based on several factors like gray level, textures, color gradient and intensity. Removal of shadow is also done by increasing the brightness of the affected region.

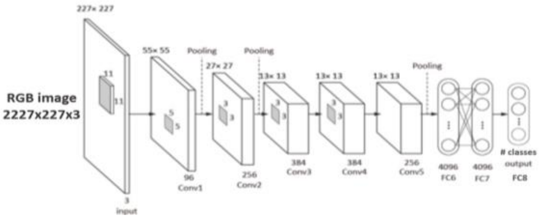


Fig. 1. AlexNet Architecture

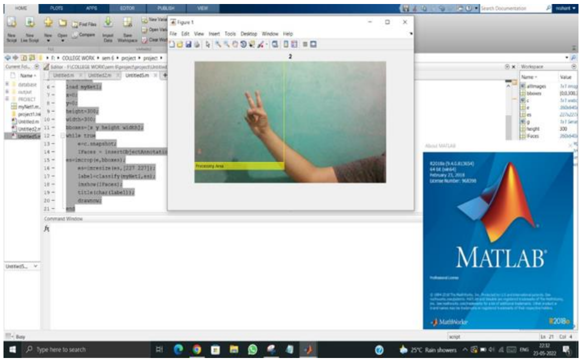
D. Features Extraction

Customized features must be extracted from the available preprocessed images. The features are extracted by applying AlexNet, a trained CNN shown in fig. 1 after segmentation and edge detection of the hand. For segmentation, k-means clustering

has been used as it provides best results when the dataset is well distinct or well separated. Edge detection is done to extract hand shape using canny edge detection. For classification of dataset, features like contour, angle of image is extracted from the detected image after the edge detection.

E. Outputs

Fig. 2 illustrates the successful implementation of hand gesture recognition on a licensed version of MATLAB. Desired output can be seen with every finger interpretation in the live feed itself.



```
load myNet1;
x=0;
y=0;
height=300;
width=300;
bboxes=[x y height width];
while true
    e=c.snapshot;
    IFaces = insertObjectAnnotatio
es=imcrop(e,bboxes);
    es=imresize(es,[227 227]);
    label=classify(myNet1,es);
    imshow(IFaces);
    title(char(label));
    drawnow;
end
```

Fig. 2. Live feed gesture recognition in MATLAB

F. Drawbacks

The given model detected the correct gesture with an accuracy of 70%. As the model was only based on 7 gestures with 300 images for each gesture. The dataset was small and could not yield better results for sign language.

Using CNN in Python (3.10.2)

A. Image Acquisition

The data is captured in frames of pictures, Webcam is used in our model for the purpose of acquisition. Further the initial image acquired is raw and not processed using filters.

B. Segmentation

In order to separate meaningful objects and signs from the given context of the acquired image the process of segmentation is used, whereby the text reduction, filtration skin color and canny edge detection are used in the preprocessing stage in fig 3 of the proposed methodology. In order to recognize the gestures, the coordinates and motion of hands are detected during segmentation.

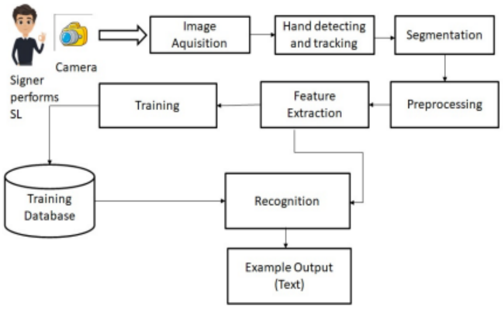


Fig. 3. Proposed Methodology.

C. Feature Extraction

Customized features are extracted from the available preprocessed images, features like form, location coordinate, angle, distance, contour, color, histogram and others from the preprocessed data, this process is necessary in order to reduce the dimensions by dividing the larger raw data into lesser and easy to handle classes so as to make the processing simpler. Since larger the data more will be the variables, and hence more computation power is required in traditional techniques, using CNN function extraction selects the best

feature from the bigger dataset which further selects and combines the larger variables into functions, which make the model simpler. Moreover, retaining its accuracy in determining the collected data.

D. Preprocessing

Preprocessing is a technique to eliminate the unwanted noise using various filters including dilation, erosion and Gaussian smoothing and others. In order to reduce the size of images they are converted to grayscale from color images.

(1) Morphological Transform

Erosion and Dilation

Dilation and Erosion are deployed for binary images, here morphological operations will create a structure feature for matching-size output images. In order to exact fit structuring with the location in a binary image, the pixels in image overlapping should be having structuring element value 1. A structuring element is said to hit at the location of the image if any of the pixels of the image overlapping the structuring element are 1. Erosion will be shrinking the objects and removing pixels from the edges of the object to a depth approximately half width of the structuring element. In Dilation, the largest value of every pixel in the adjacent will be the value of the output pixel. Pixels of a binary image are set to 1 when every of its adjacent has the value of 1; Overall, to increase the visibility of artifacts and patch up the small areas of gaps, Morphological dilation is used.

(2) Blurring

In order to eliminate the noise from any given image, blurring is used. Low Pass filters are used to diminish the unwanted noise and choose the rest of image data in original form to be used before detecting the edges of images.

(3) Thresholding

Thresholding is done in order to convert any colour or grayscale image to a binary one, into black and white. To negate the other areas which our model is not concerned with and selecting the areas of interest thresholding is used.

Fig 4 depicts the preprocessing stages of the model. In the first 60 frames the background is detected. The background with the maximum weighted average is taken and then with Each frame taken, the weighted average of each 300 frame is subtracted to obtain the foreground.

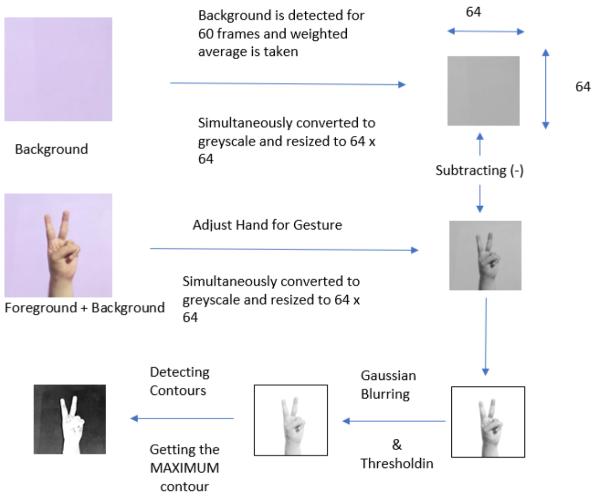

Background is detected for 60 frames and weighted average is taken
64
64
Simultaneously converted to greyscale and resized to 64 x 64
Background
Subtracting (-)
Adjust Hand for Gesture
Simultaneously converted to greyscale and resized to 64 x 64
Foreground + Background
Detecting Contours
Gaussian Blurring
& Thresholdin
Getting the MAXIMUM contour

Fig. 4. Dataset Formation and Preprocessing

E. Recognition

Here in this paper, CNN is used as a classifier. CNN's advantage is high precision, which is why it is used in image classification and recognition. The Convolutional Layer makes use of a group of learnable filters where a hierarchical model is built in the form of a network, like a funnel where output is a fully connected layer of neurons connected to one another before processing. Various other machine learning techniques were also used for identifying and understanding sign language like Principal Component Analysis (PCA), K-Nearest Neighbor classifiers, Support Vector Machine (SVM), Principal Component Analysis (PCA) and Random Forest and their performance was also compared.

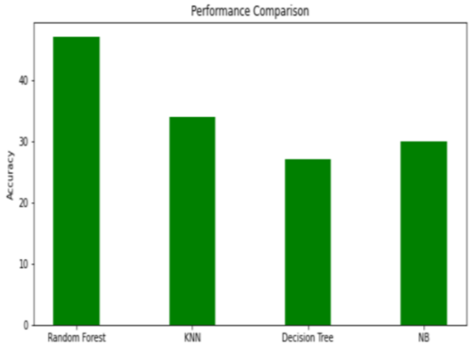


Fig. 5. Performance Comparison of Different Machine Learning Techniques

F. Output

The model was able to detect the hand correctly and also detected the gestures with higher accuracy than the above-mentioned models. The model was also able to remove unwanted things from the background by analyzing the hand contour and removing the maximum weighted average from the live video feed frame.

V. Proposed Algorithm

A. Database Creation

To create the dataset with different gestures images, need to be captured in a different background/environment. OpenCV, Numpy and Keras are the libraries used to detect hand in frames and A Region of Interest (ROI) is defined to focus only on hand during the live feed. We have captured our hand in 8 different positions (Far, Close, top left, top right, mid right, mid left, bottom right and bottom left), we have considered rotation of our hand to overcome any kind of edge case. Image Acquisition includes multiple steps such as Fetching background, detecting the hand in ROI, fixing the background and then capturing 900 frames with the given gesture in slightly different angles for all possible edge cases. For removing any unwanted thing from the background, the weighted average of background is calculated and then it is subtracted from each frame, thus minimizing the chances of any error. Each set of images is then stored in a folder. Such 20 folders were created in total to create a dataset of 24000 images.

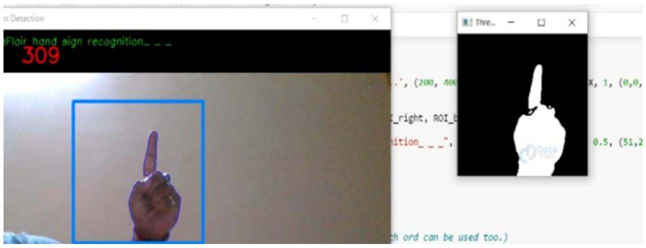


Fig. 6. Customized dataset creation using Webcam

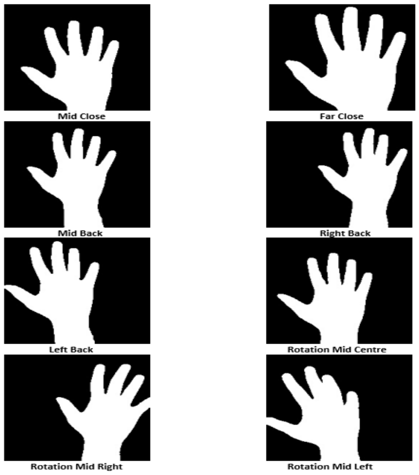


Fig. 7. Different Orientation of Hand in Dataset

B. Threshold value

Threshold value for each frame is calculated using cv.findContours function. From all frames captured, the contour with the maximum value is returned. Based on the contours, if any kind of object is detected in the foreground of the ROI i.e., hand, then the system starts to save pictures in the training dataset. The dataset is divided for training and testing into 2 sections viz testing and validation in a proportion of 8 and 2 respectively. Now, the trained model consists of 920 images for each gesture and can further validate using the other 240 images.

C. Layers of CNN model

For classification of the given pic that are static we have deployed a CNN model, our main objective behind building the neural network is to describe each of the feeded initial layers. Here every 784 pixels of an image of the 28x28 picture were shown with the gray-scale figure lying by 0 for black and 1 for white. Data has been converted into a series of numbers to become computer readable. After the preparation of input layer, the hidden layers will process them. Fig 8 shows the neural network architecture where the first hidden layer consists of many nodes, 784 of input values are their weighted total. Here the input data is passed through the rectified linear unit (ReLU), for negative it will produce the value 0 but produces the same value as input in case of positive value of data. The input of networks of any hidden layer will be the output produced by CNN layers

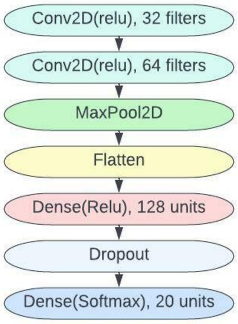


Fig. 8. Proposed CNN model architecture for the classification

D. Training CNN

Here, we have trained a CNN model on the data collected using keras ImageDataGenerator to train and test the set data by flowing from the directory function. Here labels of the given symbol folder will be the name of the class. Further to implement callbacks we have used early stopping and learning rate. For measuring the accuracy and loss after every epoch we have used a validation dataset. The LR (Learning Rate) of the model will go down when the validation loss is not decreasing and ReduceLR is used for safeguarding it from overshooting. Here the early stopping algorithm is used when the validation accuracy goes on reducing after every epoch which finally stops the training.

Epoch No.	Loss	Accuracy	Val_Loss	Val_Acc
1	12.71	0.18	0.4	0.8
2	1.44	0.53	0.3	0.88
3	0.84	0.71	0.15	0.92
4	0.56	0.81	0.08	0.94
5	0.38	0.86	0.05	0.95
6	0.30	0.88	0.03	0.95
7	0.25	0.92	0.02	0.95
8	0.27	0.95	0.01	0.95
9	0.16	0.96	0.01	0.95
10	0.15	0.98	0.01	0.96

Table 1. Accuracy as per each epoch

Here, the two algorithms used are the combination of Adagrad and RMSProp (ADAM) and the stochastic gradient descent (SGD); It will change the weighable size at the given moment of training. SDG has given _ training accuracy and _ validation accuracy to be higher and better.

E. Gesture Prediction

The focus is to detect the ROI (region of interest) for which a boundary box is generated to measure the cumulated average similar to the dataset creation process. ROI helps in detecting any kind of object in foreground i.e., hand without focusing on any other manual feature. Once the hand is detected the thresholding helps in detecting contours of the hand. The model created after training using keras.models.load_ Model is loaded with the threshold picture from the ROI as an input to the system. Upon comparison, the gesture can be predicted. The following is the result of the predictions by our model on a small test dataset that we created separately.

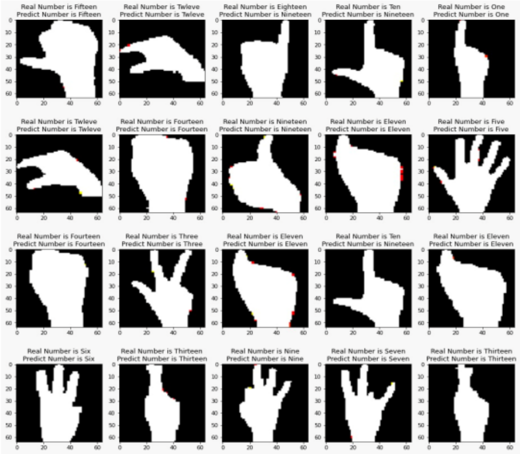


Fig. 9. Number matrix corresponding to each gesture

VI. RESULTS

When we train the model, the accuracy and loss in the model for validation data may vary depending on the instance. Loss should normally decrease as the era progresses, but exactness should increase. However, with validation loss (keras validation loss) and validation accuracy, a variety of scenarios are possible, as shown below:

1) Validation loss begins to increase, while validation accuracy decreases (starts to fade away). This implies that the model is cramming values rather than learning them.

2) Validation loss begins to decrease, while validation accuracy begins to rise. This is also fine because it means the model is learning and dealing properly. We obtained the

following findings after testing our model: we drew the graph of accuracy and loss with regard to epochs.

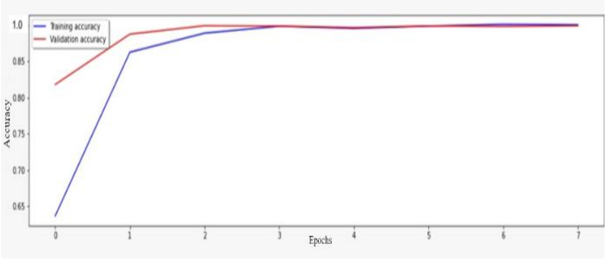


Fig. 10. CNN based Gesture detection accuracy curve

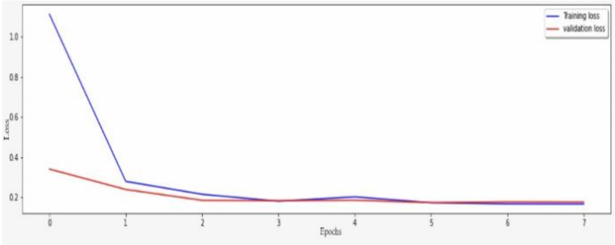


Fig. 11. CNN based gesture detection loss curve

We built a separate test set with 6430 images total of all the 20 different kinds of hand gestures to better visualize and to see how accurate our predictions are. We ran our predictions over it and created a heat map of the confusion matrix for each class corresponding to each gesture as mentioned in word_dict (i.e., 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19)

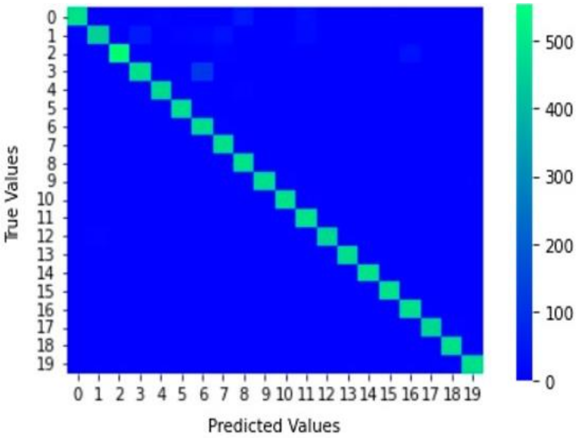


Fig. 12. Confusion matrix for predictions of Gestures

We have also released a classification report for the model that we ran over our test dataset of 6429 photos for more detailed results in terms of Accuracy, Precision and F1 score, thus a performance evaluation of the afore mentioned model. The calculations are as follows: -

	Precision	Recall	F1-score	support
0	0.97	0.80	0.88	708
1	0.98	0.73	0.84	117
2	0.98	0.92	0.95	612
3	0.89	0.79	0.83	600
4	0.97	0.95	0.96	327
5	0.95	0.99	0.97	252
6	0.76	0.98	0.86	312
7	0.88	0.99	0.93	240
8	0.88	1.00	0.93	247
9	1.00	0.99	0.99	240
10	0.97	1.00	0.98	252
11	0.90	1.00	0.94	252
12	0.99	0.95	0.97	311
13	0.98	0.98	0.98	241
14	0.99	1.00	0.99	373
15	0.98	1.00	0.99	252
16	0.93	0.99	0.96	252
17	1.00	0.98	0.99	217
18	0.97	0.99	0.98	312
19	0.98	0.99	0.99	312
Accuracy			0.94	6429
Macro avg.	0.95	0.95	0.95	6429
Weighted avg.	0.95	0.94	0.94	6429

Table 2. Performance evaluation for CNN Model

VII. CONCLUSION AND FUTURE WORK

Through this study, we presented machine learning and a CNN model for detection and classification of 20 unique hand gestures. We have trained SVM, KNN and ExtraTreesClassifier to assure that the data was sufficient to train a neural network. We then trained a convolutional neural network for the same. Our results indicate that the proposed CNN model has a good learning rate and a high accuracy of 94% in predicting 20 different classes of hand gesture (as per American Sign Language). Furthermore, as the dataset is customizable, in future dataset size and the number of gestures can be increased for better results and wider scope of the model. The results also prove that it is indeed possible to push the boundaries of existing research by focusing on maximizing results instead of diversifying into new techniques that may or may not work. We hope that our work promotes deeper research in this field, with a more vertical approach and can help towards developing an altogether new communication system for people with difficulties in speech.

REFERENCES

[1] GOI, “Census 2011 disability data,” 2022. [Online]. Available: https://www.census2011.co.in/disability.php

[2] Mahesh Kumar NB, (2018).” Conversion of sign language into Text”,International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13,Number 9 (2018) pp. 7154-7161.

[3] Kumud Tripathi, Neha Baranwal and G. C. Nandi, "Continuous Indian Sign Language Gesture Recognition and Sentence Formation", Eleventh International MultiConference on Information

Processing2015 (IMCIP-2015), Procedia Computer Science 54 (2015) 523 – 531.

[4] Noor Tubaiz, Tamer Shanableh, and Khaled Assaleh, "Glove-Based Continuous Arabic Sign Language Recognition in User-Dependent

[5] Mode," IEEE Transactions on Human-Machine Systems, Vol. 45, NO. 4, August 2015.

[6] B. Bauer,H. Heinz "Relevant features for video-based continuous sign language recognition", IEEE International Conference on Automatic Face and Gesture Recognition, 2002.

[7] Simei G. Wysoski, Marcus V. Lamar, Susumu Kuroyanagi, Akira

Iwata, (2002). "A Rotation Invariant Approach On Static-Gesture Recognition Using Boundary Histograms And Neural International

[8] Journal of Artificial Intelligence Applications (IJAIA), Vol.3, No.4, July 2012.

[9] Mokhtar M. Hasan, Pramoud K. Misra, (2011). "Brightness Factor Matching For Gesture Recognition System Using Scaled Normalization", International Journal of Computer Science Information Technology (IJCSIT), Vol. 3(2).

[10] Ashish.S.Nikam, Aarti.G.Ambedkar, "Sign language recognition using image based hand gesture recognition techniques" *International Conference on Green Engineering and Technologies (IC-GET) IEEE 2016*, e-ISBN: 978-1-5090-4556-3..

[11] A. Sharmila Konwar, B. Sagarika Borah, C. Dr.T. Tuitthung, " An American Sign Language Detection System Using HSV colour model and Edge Detection" *International Conference on Communication and Signal Processing 2014*, e-ISBN: 978-1-4799- 3358-7, INSPEC Accession no.14737547.

[12] A Sajeena, O. Sheeba, SS Ajitha, "Indian Sign Language Recognition using LexNet,". AIP Conference Proceedings 2222, 030028 (2020); https://doi.org/10.1063/5.0005665 Published Online: 16 April 2020