Real-Time Sign Language Detection Using CNN

1st Md. Nafis Saiful
dept. of CSE, BUBT
Mirpur-2, Dhaka, Bangladesh
nafissaifuldeepto@gmail.com

2nd Abdulla Al Isam
dept. of CSE, BUBT
Mirpur-2, Dhaka, Bangladesh
abdullah.isam144@gmail.com

3rd Hamim Ahmed Moon
dept. of CSE, BUBT
Mirpur-2, Dhaka, Bangladesh
moon64725@gmail.com

4th Rifa Tammana Jaman
dept. of CSE, BUBT
Mirpur-2, Dhaka, Bangladesh
rifah.chetona7@gmail.com

5th Mitul Das
dept. of CSE, BUBT
Mirpur-2, Dhaka, Bangladesh
mituldas751@gmail.com

6th Md. Raisul Alam
dept. of CSE, BUBT
Mirpur-2, Dhaka, Bangladesh
raisul@bubt.edu.bd

7th Ashifur Rahman
dept. of CSE, BUBT
Mirpur-2, Dhaka, Bangladesh
ashifurrahman.bubt@gmail.com

***Abstract*—Sign language is a system of communication using visual gestures and signs. Hearing impaired people and the deaf and dumb community use sign language as their only means of communication. Understanding sign language is so much difficult for a normal person. Therefore, the minority group has always faced many difficulties in communicating with the general population. In this research paper, we proposed a new deep learning-based approach to detect sign language, which can remove the barrier of communication between normal and deaf people. To detect real-time sign language first we prepared a dataset that contains 11 sign words. We used these sign words to train our customized CNN model. We did some preprocessing in the dataset before the training of the CNN model. In our findings, we see that the customized CNN model can achieve the highest 98.6% accuracy, 99% precision, 99% recall and 99% f1-score on the test dataset.**

***Index Terms*—Sign Language, Deep learning, CNN, Communication.**

I. Introduction

There are 15 percent of the world's population with disabilities of various kinds. A total of 466 million people is deaf, more than five percent of the population. In 2050, the population will be approximately 2.7 times larger than it was in 2000, with a projected growth of 500 million people. The speech and hearing abilities of at least 70 million people are impaired. These people deal with difficulties in interacting with others, especially when joining the workforce, education, healthcare, and transportation. The study discovered that healthcare providers failed to teach deaf women how to interact with others in a survey conducted in the United States [1]. On the other hand, the UNCRPD safeguards deaf and sign language users by guaranteeing their right to use signs [2].

Communication between the hearing and speech-impaired population and people with hearing disabilities requires interpreters as well [3]. Underprivileged and remote areas, however, are difficult to assign and train interpreters in [4], [5]. This means that those populations are lacking a vital necessity that all human beings require in order to live a normal life, just like their counterparts in developing countries, underdeveloped nations, and affluent nations [6]. There are 153,776 people with hearing disabilities, 73,507 people with visual disabilities, and 9625 people with both hearing and vision disabilities in Bangladesh, according to the Department of Social Services et al. [7]. People with hearing or speech impairments usually use sign language as their sole mode of communication. The medium of sign language is not conducive to communication between people with speech and hearing disabilities and those without. This communication barrier between people with hearing impairments and a common person can be overcome by using a digital Sign Language Interpretation system.

In this paper, we presented a new CNN-based sign language system. To detect sign language we used our customized dataset, mediapipe, OpenCV, and CNN model. Some of our major contribution to this paper is listed below,

- Building a large number of sign language word datasets.
- Recognizing sign language from hand gestures.
- Building a state-of-the-art method for detecting sign language.
- Validating our method using various evaluation matrics like confusion matrix, Accuracy, Precision, and Recall.

Here we listed all the sections contained in this paper and their descriptions. The next section contains reviews of related papers. In section III, we provided a short overview of how to set the machine for this experiment. A description of the used dataset is given in section IV. In section V, we presented a brief explanation of all the steps in our experiment, and in section VI we provided a report on the results and evaluations of our "Sign Language Detection System". In the last section, we summarize our project and our plans for future work.

II. Related works

The end of 1990 saw the first recognition of sign language.To recognize it, electrochemical devices were used as a primary method.We investigated parameters such as the position, angle, and angle of the hand using the device. Glove-based systems use this approach.Signers are required to wear a cumbersome device when using this method.Additionally, the recognition system has problems with accuracy and efficiency [8].Video clips of different gestures of sign language are analyzed in [9] and an audio expression is produced based on the analysis. There is a problem with the frame rate of

the animation in this case. The frame rate was decreased manually to make it easier to understand the sign language. It was developed to communicate with D&D personnel another system referred to as "Intelligent Assistant" [10].

With the help of a microphone, the system captured sound and converted it into text using Microsoft's Voice Command and Control Engine. Besides its inefficiency in noisy environments, this system also generated incorrect outputs due to noise included in the input [8]. Comparing to those systems [11], it is more complex.A glove with different dots on each finger was used to display the signs in this case.A real-time photo was captured of the signs using digital input.In order to understand what the sign had been shown, the program examined the dots of the graphics in the image file.Then, the wave files prerecorded as regular language signs are recognized.Using clustering, the dots were grouped together.Predefined tables were mapped to the results of this clustering.

A Bengali number range from 1 to 10 is all that was needed in this system, as no intelligent system was required to make sense of it.Sign recognition can also be accomplished using NN. Based on [12], a hybrid vocabulary of state and dynamic hand gestures was classified with Radial Basis Functions and Bayesian Classifiers.It was also shown in [13] that NN could be utilized to classify JSL.As an example, another approach proposed recently in [8] used a slow learning algorithm to recognize BdSL in cases where the test case was not properly described.

III. EXPERIMENTAL SETUP

In our research, we used google colabs to preprocess and train the CNN model. Google colabs is an online free IDE for running Python and machine learning code. We built our customized dataset for training. To use the dataset on google colabs code we need to upload all the datasets on google drive. After uploading the dataset to google drive we are ready to work on it. To preprocess the dataset and train the model we need some python and machine learning libraries and frameworks. Those libraries and frameworks are mediapipe, OpenCV, Tensorflow, Keras, Sklearn, Numpy, Pandas, etc. We can install those libraries and frameworks using the pip package of python.

IV. DATASET

There are many sign language datasets present online, but most are for English or other language number and character. In our system, we want to interpret sign language as English words. So we are building a new dataset that can help us to complete this research work easily and efficiently. The dataset we are building has a total of 11 labels. Those labels are hi, good bye, like, dislike, yes, no, happy, sad, thank you, I love you, and victory. The sample of our used dataset is given in figure 1. In our dataset, both labels have 100 images and a total of 1100 images. Those images are used to train and test the CNN model. We know the dataset is too small but our model gives a very good result on a such small dataset. In near future, we try to increase images on our used dataset.

Fig. 1. Some sample of our used dataset is given in this figure.

V. METHODOLOGY

In this paper, we proposed a new method for detecting sign language. We divided our whole system into four sub-steps. These are Data Preprocessing, Model Building, Model Training, and Real-time prediction. Al the sub-section is described here. We gave a pictorial description of all steps of our "Sign Language Detection System" in figure 2.

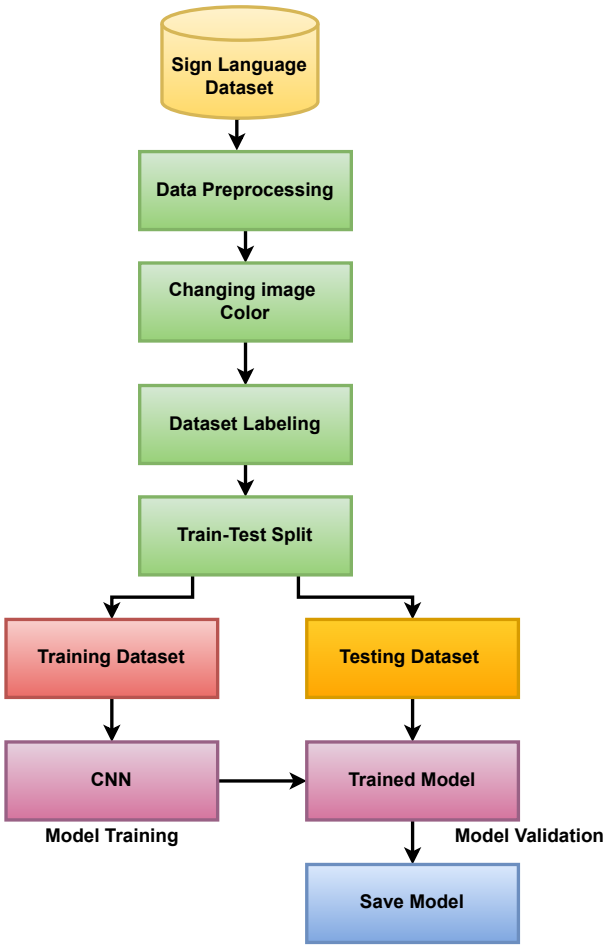

Fig. 2. This is our sign language detection system's main architecture. We can see an overview of our entire approach through this figure.

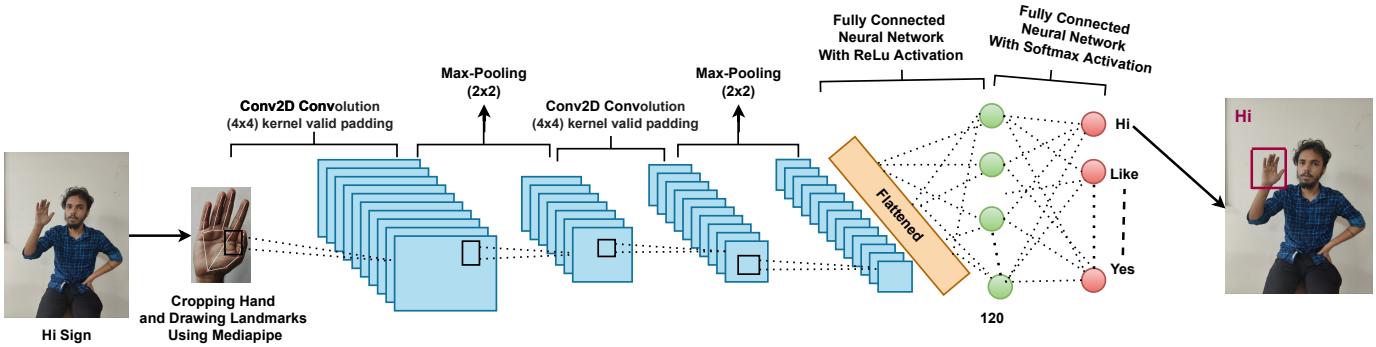

Fig. 3. In this figure, we give an overview of the model training of our sign language detection system.

A. Data Preprocessing

In the preprocessing step we preprocess all images that are present in the dataset. At first, we resize all the images into 160x120 pixels and change the color RGB image into a BGR image. After completing image resizing and coloring, now we detect the hand landmark using the python mediapipe library. Generally, mediapipe is used for detecting hand gestures and landmarks. We detected hand landmarks because we used them to detect signs. We draw hand landmarks of all the images of our dataset and save them into another dataset name as preprocess datasets. After completing preprocessing we used one hot encoding to label all the images of the dataset. After creating the label we divided dataset into 85/15 ratio to build training and testing datasets. Here training dataset is used to train the model and the testing dataset is used to validate the model.

B. Model Building

In our experiment, we used the CNN model to detect sign language. We used CNN because it gives very good

accuracy on the classification tasks. We used the Conv2D layer of CNN to detect sign language. In the first layers of our Conv2D, we set the input shape as 120x160x3 because our converted image has RGB color with a size of 120x160 pixels. We set the filter size as 32, kernel size 4x4, and activation function as ReLu in the first Conv2D layer. After that, we dropped 0.5% of neurons in the first layer using the dropout layer. In the next layer, we added a pooling layer. In the pooling layer, we used max pooling with a pool size of 2x2.In the next layer, we added another Conv2D layer with a filter size of 64, kernel size of 4x4, and activation function as ReLu.Followed by the Conv2D layer we added the same dropout and pooling layer that was already described above. After that, we flatten all the 2d layers into 1d layers and add two dense layers in the last of our model. In the last dense layer, we add 11 neurons and a softmax activation function. We set 11 neurons in the last layers because we have only 11 labels on the dataset and the softmax activation function is used here because our classification task is a multi-class classification task. The softmax activation function is used when we need to classify more than two classes.

C. Model Training

For model training at first, we need to compile the CNN model that was described above. To compile the model we used compile function and set the loss function as 'categorical_crossentropy', optimizer as 'adam', and metrics as 'accuracy'. After completing the compilation of the model we are ready to train it using the training dataset. For training, we used the fit method and train the model into 20 epochs. We got a very good accuracy after training the model in 20 epochs. We try with some other number of epochs but we think a 20 epoch run is best for our model so we choose 20 epochs as best. In figure 3 we gave a simple prototype of our "Sign Language Detection System".

D. Real-time Prediction

After fully training the CNN model with our customized preprocess dataset we save the model for real-time prediction. In the real-time prediction step, we draw the hand landmarks and analyze the hand position with the mediapipe and give them into the trained model. Now the trained model finds the best match between the given sign and the dataset sign. The model predicts the sign that matches best for the given sign. In figure 4 we given some of the sample real-time prediction images of our CNN model.

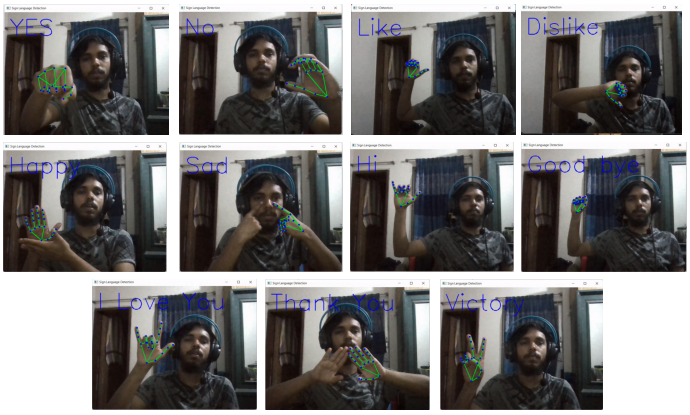

Fig. 4. Some real-time prediction result of our trained model is given in this figure.

VI. RESULT ANALYSIS

In this section, we give all the results and evaluation metrics of our "Sign Language Detection System".

A. Evaluation Metrics

1) Confusion Matrix: Confusion matrices are tabular representations of the prediction model's performance. In a confusion matrix, each entry represents the number of predictions made by the model that was correct or incorrect. Our problem is a multi-class classification problem, so we need a multi-class confusion matrix to evaluate the model. The multi-class classification confusion matrix does not contain any positive or negative classes, like binary classification. Due to the lack of positive or negative classes, finding TP, TN, FP, and FN may be a little challenging. Here we were given a simple calculation of all the TP, TN, FP, and FN for the class "YES".

$$TP = 32, FP = 2, TN = 0, FN = 0 \tag{1}$$

Using the confusion matrix, we now measure some of the most common performance metrics for the "YES" class.

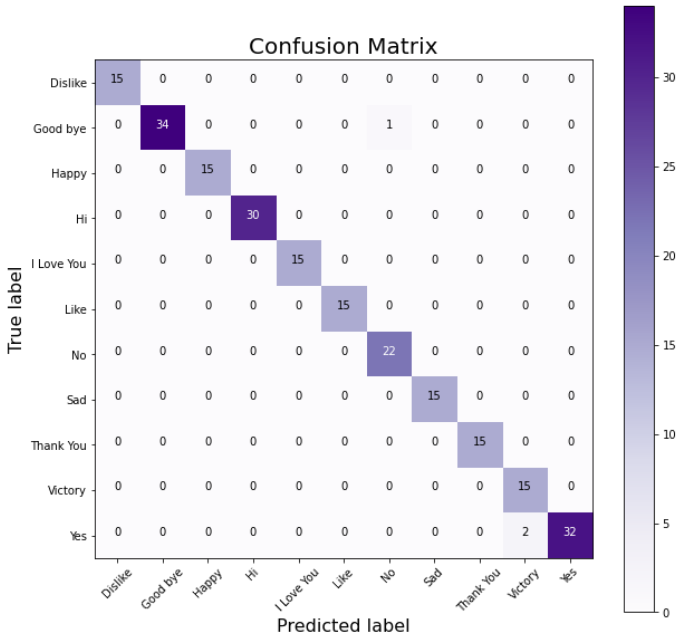

Fig. 5. The confusion matrix of our sign language detection system is given in this figure.

Accuracy: Classifier accuracy is measured by the fraction of samples correctly classified by the model. Here is the formula for calculating accuracy:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{2}$$

Precision: It tells you the proportion of positive predictions that were actually correct.Here is the formula for calculating precision:

$$Precision = \frac{TP}{(TP + FP)} \tag{3}$$

Recall: Recall indicates the percentage of positive samples that were correctly classified as positive by the classifier.Here is the formula for calculating recall:

$$Rcall = \frac{TP}{(TP + FN)} \tag{4}$$

2) Accuracy & Loss Curve: The performance of a deep learning model can be illustrated by accuracy and loss curves. The following graph shows how deep learning models improve their performance per epoch. In order to plot accuracy and loss curves, we place accuracy and loss on the y-axis and epoch on the x-axis. Our CNN model was trained over 20 epochs. Using this curve, we can determine if a model is overfitted or underfitted. We gave our model's accuracy and loss curves in figure 6.

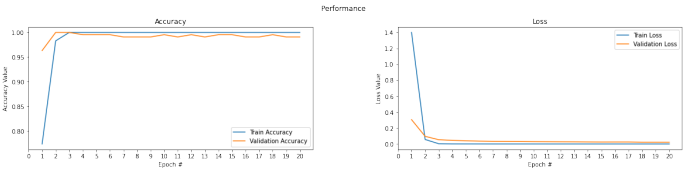

Fig. 6. We see the accuracy and loss curves of our sign language detection system as they relate to the epoch in this figure. This figure shows accuracy and loss on the y-axis and epoch on the x-axis.

B. Result

In this experiment, we detected English sign language using a Convolutional Neural Network (CNN). For this research, we built a large number of sign language words to train our model. After training the model we tested our model using the testing dataset. The result of our sign language detection system is given in table I. After testing the model, if the model gives very good results we save it for making a real-time prediction system. Some of the real-time prediction of our trained model is given in figure 4. We clearly see that our model classified all the sign language very accurately and efficiently.

TABLE I
IN THIS TABLE, WE GIVEN OUR SIGN LANGUAGE DETECTION SYSTEM ALL RESULTS.

Architecture	Accuracy	Precision	Recall	F1-Score
CNN	**98.6%**	**99%**	**99%**	**99%**

VII. CONCLUSION

Detecting sign language has become an important research field to improve communication with deaf and dumb people. It is also important to understand that different sign languages are developed in different language communities, and research on sign language detection is also language-specific. Even though English is a mainstream language with a large deaf and dumb community, there has been very little research conducted on sign language detection in English. In this paper, we propose a new English sign language detection scheme that relies on fingertip position as a training tool for a CNN

classifier. Several methods have been tested and compared against a large dataset of images. Based on test set accuracy, the proposed method outperforms all existing methods. In addition, the proposed scheme appears to be compact and efficient in terms of computation and size.

REFERENCES

[1] J. Ubido, J. Huntington, and D. Warburton, "Inequalities in access to healthcare faced by women who are deaf," *Health & Social Care in the Community*, vol. 10, no. 4, pp. 247–253, 2002.

[2] A. Lawson, "United nations convention on the rights of persons with disabilities (crpd)," in *International and European Labour Law*. Nomos Verlagsgesellschaft mbH & Co. KG, 2018, pp. 455–461.

[3] H. Haualand and C. Allen, *Deaf people and human rights*. World Federation of the Deaf, 2009.

[4] J. Napier, "Sign language interpreter training, testing, and accreditation: An international comparison," *American Annals of the Deaf*, vol. 149, no. 4, pp. 350–359, 2004.

[5] C. C. Yarger, "Educational interpreting: Understanding the rural experience," *American Annals of the Deaf*, vol. 146, no. 1, pp. 16–30, 2001.

[6] B. O. Olusanya, R. J. Ruben, and A. Parving, "Reducing the burden of communication disorders in the developing world: an opportunity for the millennium development project," *Jama*, vol. 296, no. 4, pp. 441–444, 2006.

[7] D. of Social Services, "Ministry of social welfare, g.o.p.r.o.b. disability information system." https: //www.dis.gov.bd, [Online; accessed 26-August-2022].

[8] F. M. Rahim, T. E. Mursalin, and N. Sultana, "Intelligent sign language verification system–using image processing, clustering and neural network concepts," *International Journal of Engineering Computer Science and Mathematics*, vol. 1, no. 1, pp. 43–56, 2010.

[9] D. S. H. Pavel, T. Mustafiz, A. I. Sarkar, and M. Rokonuzzaman, "Geometrical model based hand gesture recognition for interpreting bengali sign language using computer vision," in *ICCIT*, 2003.

[10] A. Eshaque, T. Hamid, S. Rahman, and M. Rokonuzzaman, "A novel concept of 3d animation based'intelligent assistant'for deaf people: for understanding bengali expressions," in *ICCIT*, 2002.

[11] S. Rahman, N. Fatema, and M. Rokonuzzaman, "Intelligent assistants for speech impaired people," in *ICCIT*, 2002.

[12] A. Sandberg, "Gesture recognition using neural networks," *Praca magisterska*, 1997.

[13] K. Murakami and H. Taguchi, "Gesture recognition using recurrent neural networks," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1991, pp. 237–242.