

# American Sign Language Alphabets Recognition using Convolutional Neural Network

Diponkor Bala<sup>1\*</sup>, Bappa Sarkar<sup>2</sup>, Md. Ibrahim Abdullah<sup>3</sup> and Mohammad Alamgir Hossain<sup>4</sup>

<sup>1</sup>Department of Computer Science and Engineering, Islamic University, Kushtia, Bangladesh.  
Email: diponkor.b@gmail.com

<sup>2</sup>Department of Computer Science and Engineering, Islamic University, Kushtia, Bangladesh.  
Email: bappa@cse.iu.ac.bd

<sup>3</sup>Department of Computer Science and Engineering, Islamic University, Kushtia, Bangladesh.  
Email: ibrahim@cse.iu.ac.bd

<sup>4</sup>Department of Computer Science and Engineering, Islamic University, Kushtia, Bangladesh.  
Email: alamgir@cse.iu.ac.bd

\*Corresponding Author

**Abstract:** Sign language is a kind of language which is primary used for the deaf and hard of hearing and also used for those who are unable to physically speak. Sign language is only understood by a small percentage of the population. There is a huge communication gap between the deaf community and the hearing majority but it is not acceptable for a nation. Computerized sign language recognition capabilities attempt to break through this communication gap. Due to the advancement of technology, Artificial Intelligence has made human life easier by using various machine learning techniques. Computer vision is one of the most vital fields of artificial intelligence. However, utilizing computer vision to recognize American Sign Language is extremely difficult since sign language is extremely complicated and has a large inter-class variation. In the last few years, Convolutional Neural Network has become an effective way for the classification of multiclass images. In this paper, we have used Convolutional Neural Network (CNN) for the recognition of ASL alphabets. In this study, we have used the Sign Language MNIST dataset which consists of 34627 images where 27,455 training samples and 7172 testing data. The dataset includes 24 alphabets except for the letter J and Z among 26 alphabets. For the recognition of ASL alphabets, at first, we pre-processed our dataset by the normalization technique. After that, we designed a convolutional neural network architecture for extracting features from hand gesture images and then we trained our CNN model through the training dataset. We have finally evaluated our proposed CNN model based on the test dataset and obtained its accuracy to see that how many ASL alphabets recognize correctly. The proposed CNN architecture was able to achieve an accuracy of 99.78% on unseen data.

**Keywords:** American sign language, ASL alphabet, Convolution Neural Network (CNN), Hand gesture, Recognition.

## I. INTRODUCTION

The Sign Language alphabets are generated by employing hand and facial movements to transmit the thoughts of persons with hearing and speech difficulties to others who do not have these difficulties. Everyone who want to communicate with the deaf has to use sign language. Sign language is an important way to communicate not only between the deaf but also between the deaf and the non-deaf. Most of the population may not purely understand Sign language. As a result, there is a significant communication gap between normal individuals and those with impairments [1]. As it is difficult for any human to be with a handicapped person all of the time and aid them in their everyday activities as a translator, various technological developments have been made to assist individuals in speaking and hearing-impaired persons. A strategy that uses technology to transform sign language into interpretable language that is understood by the general public might be considered. This method will undoubtedly narrow the communication barrier and make communication between impaired and non-impaired persons more practicable [2]. Over a hundred distinct sign languages, such the Indian Language, the American Sign Language (ASL), the Russian Sign Language, etc., are used across the world by groups with a disability. Sign Language is their medium of communication, used by hundreds and millions of individuals worldwide and approximately millions in India.

American Sign Language is widely utilized and ranks in North America as the fourth most commonly used language [3]. It

may surprise you to learn that, in addition to the United States, ASL is used in more than 30 additional countries where English is the primary language of communication. ASL is used as the primary form of communication by around a million individuals in the United States and other areas of the world [4]. ASL is a large and complicated language that uses hand, finger, and facial movements to represent the ideas of persons who are deaf. The American Sign Language (ASL) is acknowledged as an adequate and unique language. It has a multitude of variants, much like any other language, such as Italian and German. Sign Language is a great work of interactive art that benefits a large group of individuals who are deaf or hearing disabilities. The availability of ASL increases enjoyment, hope, and widespread influence among disabled persons.

ASL is a set of 26 gesture symbols that may be used for expressing various English Dictionary terms, known as the American Manual Alphabet. There are 19 various hand forms developed by people to convey 26 manual alphabets in ASL. Since there are fewer hand forms accessible, changing the orientation of a hand shape allows us to transmit multiple alphabets. For instance, the letters 'K' and 'P'. Hand gestures may also be used to convey numerals ranging from '0' to '9'. Although there are no hand gestures for representing individual words or concepts, there are a variety of facial and hand gesture indications that may be used to communicate different English words. Fig. 1 shows a collection of 26 alphabets of signs of English alphabets samples from A to Z.

With the development of technology, computer vision has become an effective way to provide help in people's daily life. So many applications have been developed in almost every field of industry, education, agriculture and business. As the hardware has been improving increasingly, there are lots of intelligent algorithms coming out, like Support Vector Machine [5], Deep Neural Network, which have been used in the applications of smart home, e-learning, robot and so on. Deep Neural Network (DNN) [6] is a highly effective way for image classification, especially convolutional neural network which is also called CNN. The structure of Convolutional Neural Network [7] consists of Input layer, several pairs of convolutional layers and pooling layers and output layer. In each convolutional layer,  $n \times n$  matrix is used to get convolved features. The aim of this is to get high level feature such as edges. The pooling layer returns the maximum value or average value from the portion of the image covered by the matrix. We have used Convolutional Neural Network architecture to recognize the American Sign Language alphabets and achieve a high accuracy on unseen data.



Fig. 1: Some Sample ASL Alphabet Images of Dataset

The article is structured as follows: The introduction to this study is included in Section I, Section II includes a description of related works, Section III contain the brief description of the dataset, the brief description of the proposed methodology given in Section IV, the essential description of evaluation metrics that we have used for the analysis of the performance of the proposed model is given in Section V, Section VI describes results and discussion about this work and finally, in Section VII, the conclusion of this research work has been drawn.

## II. RELATED WORKS

From the last decades, many researcher has been focusing on the recognition of sign languages from numerical and image data through various methods. Some previous work has been accomplished by Support Vector Machine (SVM) classification algorithm [5]. SVM has been utilized for the early classification of gestures in static ASL and parallel Hidden Markov models [8] have also been utilized. The description of the some exists work is given below:

Sharmila Konwar *et al.* [9] presented an American Sign Language detection system based on HSV color model that is used to detect hand gestures color and edge detection technique that is used to detect the shape of hand gestures from the images. They have obtained 65% ASL recognition rate. Sruthi Upendran *et al.* [10] proposed a scheme to recognize and interpret ASL alphabets. They have used the concept of PCA to extract features from images and KNN classifier to classify hand gestures. The proposed system accuracy rate is 77.29%.

Vivek Bheda *et al.* [11] proposed a method to recognize ASL digits and alphabets based on deep convolutional neural network architecture. They have achieved 82.5% and 97% accuracy on alphabet gestures and digits respectively. Srinath *et al.* [12] presented a segmentation technique based classification approach for the recognition of sign language. The proposed model recognizes ASL alphabets gestures with a success rate of 86.67%.

Jayshree *et al.* [13] proposed a hand gesture recognition system for the recognition of ASL alphabets based on edge orientation histogram technique. They have achieved 88.26% accuracy for the recognition of ASL alphabets. Joyeeta Singha *et al.* [14] developed a dataset of 240 images with 24 indigenous signs or ISL language. The Eigen values were then retrieved and categorized according to their Euclidean distance from the images in the data set. 97% accuracy was achieved by this technique.

G. A. Rao *et al.* [15] proposed a Convolutional Neural Network (CNN) architecture for the recognition of sign language and achieved 92.88% recognition rate. Dhruv Rathi *et al.* [16] recommended a system that is able to recognize the alphabets of ASL based on Transfer Learning approach. He used MobileNet and InceptionV3 model and achieved 95.06% and 93.36% respectively.

### III. DATASET

One of the key and critical components of every investigation is data collection. Data that are relevant as well as satisfy the parameters for study must be collected. The data must include readings reproducing all the possibilities, even severe situations. This serves to make greater precise and specific observations. The acquired data is the basis of any research and should be carefully collected.

We have opted a dataset named Sign Language MNIST for our research. Sign language MNIST [17] is a dataset used by researcher who wants to build the model for sign language alphabet classification. It comes from the MNIST database (Modified National Institute of Standards and Technology database) which is a large database of handwritten digits used for image-based machine learning methods. Same with MNIST, Sign language MNIST has 28×28 pixel (784 total) per sample. Instead of black and white image in MNIST, the image in sign language MNIST are all colorful. It does not include all 26 alphabets, for the letter J and Z requires motion. The dataset consists of 34627 images where 27,455 training samples and 7172 testing data. This is approximately half the size of the MNSIT dataset. Each image has a label (0-8, 10-24). There is no 9 and no 25, for we exclude the letter J and letter Z. The all images in dataset are saved in csv file. Each row represents a single image. The first column is label (the number of class that this image is). From the 2<sup>nd</sup> column to 785<sup>th</sup> column, there are

784 pixels. Each pixel has grayscale values between 0-255 for the image is converted to gray. These images show multiple persons repeating the gesture against different backgrounds. After the data set has been explored, the number of training and testing samples for each level is shown in Fig. 2.

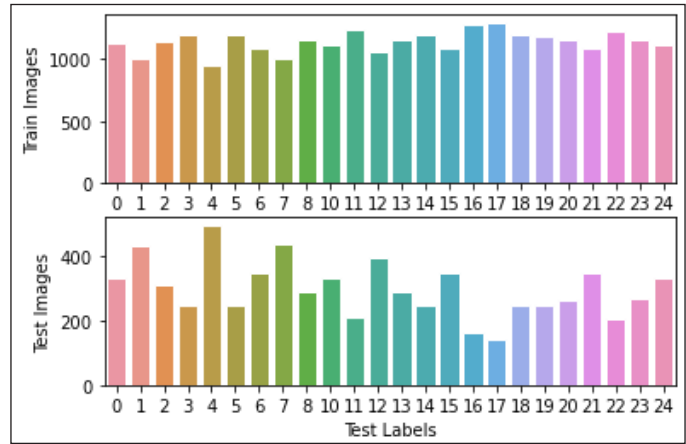


Fig. 2: Number of Training and Testing Samples for Each Label

### IV. METHODOLOGY

The proposed methodological steps for the recognition of American Sign Language alphabets is shown in Fig. 3.

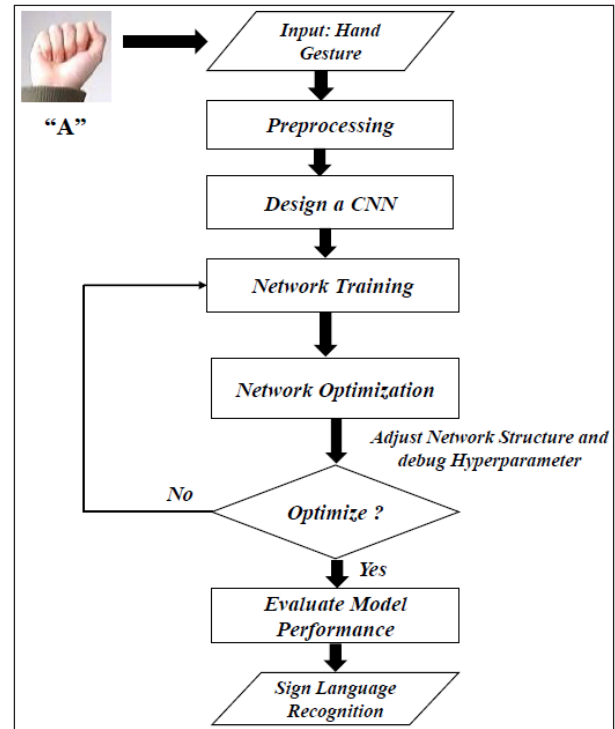


Fig. 3: Methodological Steps for the Recognition of American Sign Language Alphabets

### A. Loading and Pre-Processing of Dataset

We actually started first loading the dataset for working with and modifying them so that they could be used by various classifiers to accurately predict the sign language alphabet. We have already different training and testing sets in the supplied dataset. We viewed our dataset and observed that it was balanced. In order to eliminate the effects of lighting variances, we carried out a grayscale normalization technique [18]. Some transformed grayscale images is shown in Fig. 3. In addition, on [0...1] data the CNN computes quickly and efficiently than on [0...255]. We used the following Eqn. 1 to normalize the image. If the normalized value is represented by  $x'$  then

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

where  $x$  is the original intensity of the image.

We divided the training set into two parts after the normalization: the training set and the validation set, which included 80% and 20% of the training set, respectively. Each image in the dataset has 784 columns. By turning every row of 784 columns into a (28, 28, 1) matrix, we were able to input this data into the proposed CNN model.

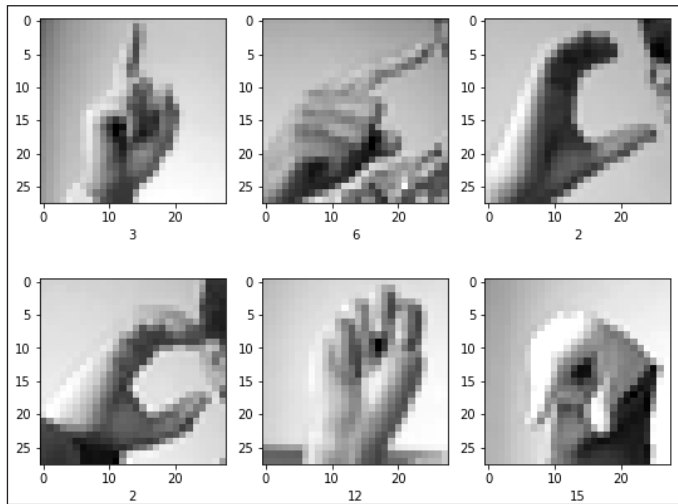


Fig. 4: Grayscale Images

### B. Data Augmentation

Augmentation [19] describes the process of extending the size of the dataset used for training. As a result, we may have more variability in our dataset, and it is also utilized to expand the number of training samples in order to prevent overfitting. Our data augmentation involves random featurewise and samplewise center, random featurewise and samplewise std normalization, random zca whitening, random rotation ranging 10 degrees and random height and width shift in the range of 0.1, random zooming, and random flips.

### C. Convolutional Neural Network Model Description

We proposed a Convolutional Neural Network architecture [7] [20] for the recognition of ASL alphabets, which comprises of eight convolutional layers, three pooling layers, two dropout layers and a fully connected layer. The key layers of the proposed CNN model is shown in Fig. 5 and then described below:

*Input Layer:* Through this layer, the model propagates information from the input layer to the hidden layer and then to the output layer. The shape of the image for the input layer in this study is (28, 28, 1), where the height and width of the image are 28 and one channel.

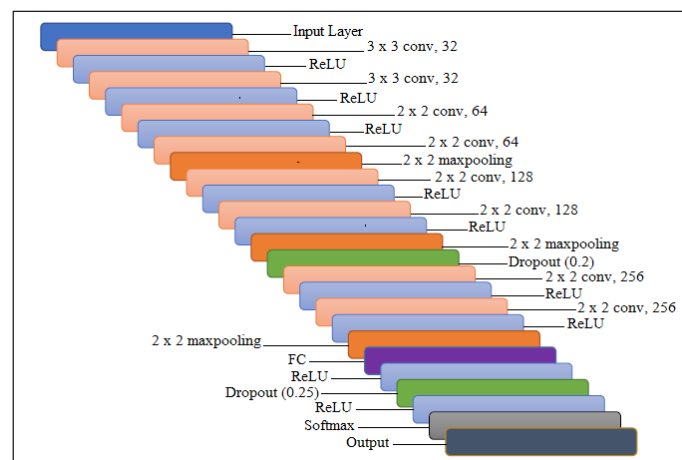


Fig. 5: The Proposed CNN Architecture

*Convolutional Layer:* Our CNN model makes use of a mathematical procedure known as convolution. The fundamental goal of the convolution operation is to use a feature extraction technique to explore features in our images and place those in a feature vector [7]. This convolution operation is executed by using two dimensional matrix ( $x$ ) known as kernel, which scans all the data matrix available. Here let's take an input image  $I$  convoluted by  $k$ , the operation is expressed as:

$$(I * k)_{ij} = \sum_{m=-\frac{k_1}{2}}^{\frac{k_1}{2}} \sum_{n=-\frac{k_2}{2}}^{\frac{k_2}{2}} I(i-m, j-n)k(m, n) \quad (2)$$

In this study, the feature extractors are composed of cov3x3, 32; cov3x3, 32; cov2x2, 64; cov2x2, 64; cov2x2, 128; cov2x2, 128; cov2x2, 256 and cov2x2, 256. The size of feature maps for the convolution operation are 26x26x32, 24x24x32, 23x23x64, 22x22x64, 10x10x128, 9x9x128, 3x3x256 and 2x2x256.

*Activation Layer:* The ReLU activation function [21] is applied in CNN models to enhance non-linearity. At the Convolution layers, the ReLU activation function was applied. If the value is greater than zero, the input is passed; else, it returns 0. The mathematical formulation of the ReLU activation function is as follows:

$$\text{ReLU}(x) = \max(0, x) \quad (3)$$

where  $x$  is the neuron's input value.

*Pooling Layer:* The pooling layer helps to minimize overfitting by eliminating a few parameters while executing a pooling operation to obtain a pooled matrix. Downsampling is another term for pooling. Max pooling [22] is a technique for focusing just on the necessary information in an image by extracting mainly greater yields from just a part of the input by using an empty feature extraction technique and also decreases the dimensionality of the feature matrix layouts, which accelerates network time. In this study, the size of the feature maps for pooling operation are  $11 \times 11 \times 64$ ,  $4 \times 4 \times 128$  and  $1 \times 1 \times 256$ . The mathematical expression of maxpooling operation can be defined as:

$$N_{out} = \text{floor}\left(\frac{N_{in} - F}{S}\right) + 1 \quad (4)$$

where  $N_{in}$  = The size of the input image

$F$  = The kernel size

and  $S$  = The stride size

*Dropout Layer:* The dropout layer [23] is utilized after convolution layers to prevent overfitting. In proposed CNN, we applied two dropout layers with dropping rate of 20% and 25% respectively, which indicates that 20% and 25% of random neurons are turned off to avoid overfitting of our model.

*Fully Connected Layer:* After layers of the convolution and max pooling, we obtain a set of feature cards. We connect them into one vector and this vector will be fed into the fully connected network. Fully connected layer [23] is utilized to give the result of classification. In this study, we carried out a flattening procedure and it'll be transformed into a  $256 \times 1$  vector and sent to the dense layer along with some predefined parameters.

*Output Layer:* The final predicted class is achieved through this layer. The sigmoid function is the core of the probabilistic technique, and it performs best when classifying between two classes. In order to the multiclass classification soft maxfunction is used and softmax [24] makes sure that the sum of the probability of the outcome is one. For this reason, we have used is the Softmax activation function for the proposed CNN. The linear input data is transformed into a probability array of all 25 classes via softmax, and the probability of occurrence is tested against the actual output. The softmax function is defined as:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}; \text{ for } i = 0, 1, \dots, K \quad (5)$$

This function takes a  $K$ -dimensional input vector and returns a  $K$ -dimensional vector of values within the range 0 to 1 that sum to 1.

#### D. Network Compiling and Training

After successfully build the CNN model, it is ready to compile. For the calculation of the model loss, we have used categorical

cross entropy loss function [25]. We have use stochastic Gradient Descent method [26] based adaptive moment estimation (Adam) [27] optimizer as our optimization method to quickly and efficiently reach convergence of our model. The total trainable parameters to be calculated are 795,640 and there are no non-trainable parameters in the proposed CNN architecture.

After successfully compile the CNN model, the model was trained. During training the model, the hyperparameters [28] like learning rate, batch size and number of epochs are performed a great role. We have set the value learning rate is 0.001, batch size is 250 and the number of epoch is up to 100. We have used a callback to determine if our validation set accuracy did not increase even after 2 (patience level) consecutive epochs during the model fitting stage. Then, we will alter our learning rate by a factor of 0.5. The training dataset used for training our model and validation dataset used for validating our model. We performed our experiment on compiler Jupiter Notebook integrates with some numbers of packages with an "Intel® Core™ i7-10510U (1.8 GHz, up to 4.9 GHz, 8 MB cache, 4 cores) powered with NVIDIA GeForce MX330 (2GB), 16GB DDR4 RAM" and Windows 10 based 64 bit operating system. We have used Google Colabratory for training and testing data. After that, we examined our proposed model is based on predictions made by our trained model on the test dataset and obtained its accuracy, precision, recall, and F1-score to evaluate the proposed model's performance.

## V. PERFORMANCE METRICS

To estimate the performance of the proposed model, four different measurements which are accuracy, precision, recall, and F1 score are utilized to measure the performance of the model. The confusion matrix can calculate these four measurements. Definition of the four measurements and confusion matrix are illustrated below:

### A. Confusion Matrix

The confusion matrix [29] is utilized to illustrate the performance for machine learning classification problem whose output can be two or more classes. Pneumonia classification is a multiple classification problem which is suitable to utilize confusion matrix to measure performance. Confusion matrix includes four significant parts which are true positive (TP), true negative (TN), false negative (FN) and false positive (FP). This matrix can be utilized to measure accuracy, recall, precision and F1 score.

### B. Precision

Precision [29] is proposed to illustrate the percentage of images which are actually positive, out of the positive classes which are predicted correctly. It can be defined as:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

C. Recall

The recall [29] is utilized to illustrate the percentage of images which are predicted correctly, out of the positive classes. It can be defined as:

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

D. Accuracy

Accuracy [29] is a common measure in machine learning and is defined as the ratio of correctly classified samples to all samples in the dataset. It can be defined as:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{8}$$

E. F1 Score

The F1-score [29] (sometimes known as the balanced F-beta score) is a single metric that combines both precision and recall via their harmonic mean. It can be defines as:

$$F1 = \frac{2 * Recall * Precision}{Recall + Precision} \tag{9}$$

VI. RESULTS AND DISCUSSION

After training, the network was tested on the testing dataset, which included 7,712 samples that had not been seen by the network during training. Performance measures including as accuracy, precision, recall, and F1 scores were measured after it was run on the testing dataset, the results are summarized in Table I.

TABLE I: PERFORMANCE OF THE PROPOSED CNN MODEL

	Precision	Recall	F1 Score	Test Accuracy
Result	1.00	1.00	1.00	0.9978

Fig. 6 and Fig. 7 illustrate the accuracy and loss graphs for training and validation accuracy and loss, respectively.

In addition to the computed performance metrics presented in Table I, a confusion matrix was also obtained. A confusion matrix shows the true positive and negative effects after the model has been executed. This provides us a clear understanding of the faulty model and the number of true negatives or false positives. The confusion matrix helps in deducing that most of the predictions made by the model are correct. The confusion matrix is shown in Fig. 8. However, due to the similarity of several letters, significant errors might be noticed. For example, the location of the fingers for A and S is fairly similar, resulting in some incorrect predictions.

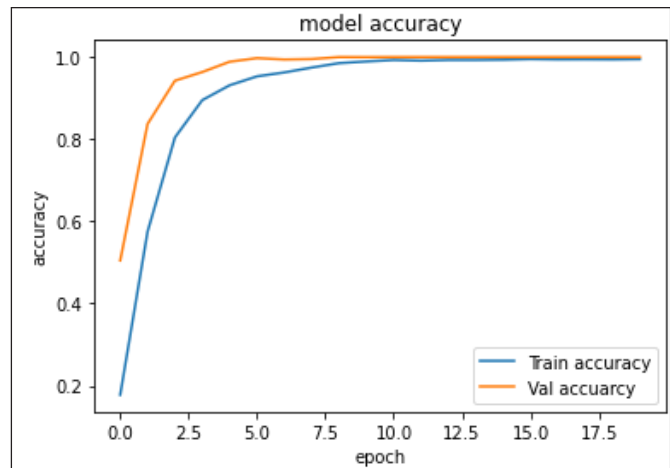


Fig. 6: Model Accuracy

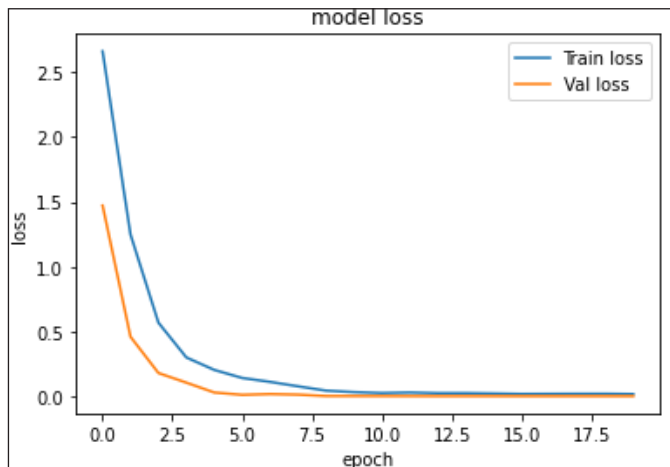


Fig. 7: Model Loss

The Classification report obtained at the end is shown in Fig. 9 and precision, Recall, F1 Score for each label is also shown.

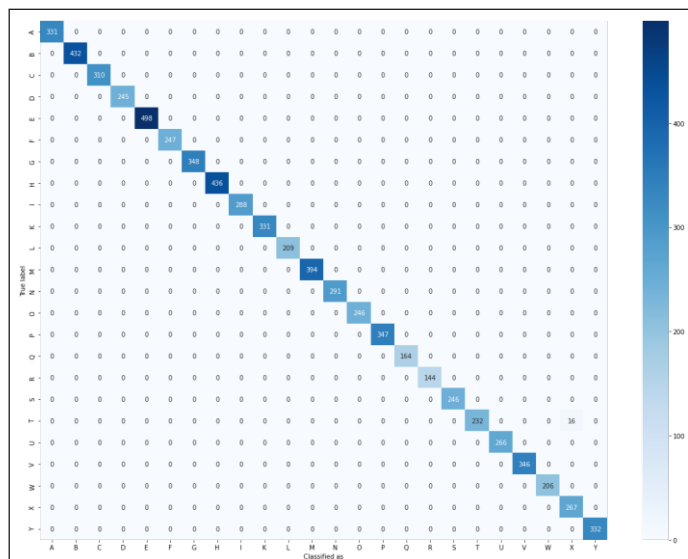


Fig. 8: Confusion Matrix

	precision	recall	f1-score	support
A	1.00	1.00	1.00	331
B	1.00	1.00	1.00	432
C	1.00	1.00	1.00	310
D	1.00	1.00	1.00	245
E	1.00	1.00	1.00	498
F	1.00	1.00	1.00	247
G	1.00	1.00	1.00	348
H	1.00	1.00	1.00	436
I	1.00	1.00	1.00	288
K	1.00	1.00	1.00	331
L	1.00	1.00	1.00	209
M	1.00	1.00	1.00	394
N	1.00	1.00	1.00	291
O	1.00	1.00	1.00	246
P	1.00	1.00	1.00	347
Q	1.00	1.00	1.00	164
R	1.00	1.00	1.00	144
S	1.00	1.00	1.00	246
T	1.00	0.94	0.97	248
U	1.00	1.00	1.00	266
V	1.00	1.00	1.00	346
W	1.00	1.00	1.00	206
X	0.94	1.00	0.97	267
Y	1.00	1.00	1.00	332
accuracy			1.00	7172
macro avg	1.00	1.00	1.00	7172
weighted avg	1.00	1.00	1.00	7172

Fig. 9: Classification Report

## VII. CONCLUSION

As interest in computer vision grows, so does its wider use and application. The number of people who use sign language is certainly significant and reading it with computer vision could be made easier for people who use that language for communication. American Sign Language is a very popular sign language all over the world. This research presents a Convolutional Neural Network architecture-based method for the recognition of American Sign Language alphabets. Our model differs from others that depend mostly on the transfer learning approach. Because of the hand movement similarity, letters like 'J' and 'Z' were excluded from the classification. Depended on the proposed method, the result showed excellent classification performance. On our test dataset, the suggested CNN model achieved an accuracy of 99.78%. The proposed CNN model is also applicable for the recognition in other sign languages and also can be used for the application with hardware based sign language recognition system.

In the future, this work can certainly be upgraded. Recognition of the alphabet is just the first step. By extending the field of view from the palm/hand to the posture of the body and head, and the facial expression could be upgraded to words, which will make communication even easier.

## ACKNOWLEDGMENT

We would like to express our heartfelt gratitude to every one of the Computer Science and Engineering department teachers in Islamic University, Bangladesh and most importantly our beloved parents for their continuous support.

## REFERENCES

- [1] M. E. Al-Ahdal, and M. T. Nooritawati, "Review in sign language recognition systems," *2012 IEEE Symposium on Computers & Informatics (ISCI)*, 2012, pp. 52-57, doi: 10.1109/ISCI.2012.6222666.
- [2] K. Nimisha, and A. Jacob, "A brief review of the recent trends in sign language recognition," *2020 International Conference on Communication and Signal Processing (ICCCSP)*, 2020, pp. 186-190, doi: 10.1109/ICCCSP48568.2020.9182351.
- [3] S. Shivashankara, and S. Srinath, "A review on vision based American sign language recognition, its techniques, and outcomes," *2017 7th International Conference on Communication Systems and Network Technologies (CSNT)*, 2017, pp. 293-299, doi: 10.1109/CSNT.2017.8418554.
- [4] S. Y. Kim, H. G. Han, J. W. Kim, S. Lee, and T. W. Kim, "A hand gesture recognition sensor using reflected impulses," in *IEEE Sensors Journal*, vol. 17, no. 10, pp. 2975-2976, May 15, 2017, doi: 10.1109/JSEN.2017.2679220.
- [5] S. Ghosh, A. Dasgupta, and A. Swetapadma, "A study on support vector machine based linear and non-linear pattern classification," *2019 International Conference on Intelligent Sustainable Systems (ICISS)*, 2019, pp. 24-28, doi: 10.1109/ISS1.2019.8908018.
- [6] H. Yi, S. Shiyu, D. Xiusheng, and C. Zhigang, "A study on deep neural networks framework," *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, 2016, pp. 1519-1522, doi: 10.1109/IMCEC.2016.7867471.
- [7] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.
- [8] C. Vogler, and D. Metaxas, "Parallel hidden Markov models for American sign language recognition," *Proceedings of the 7th IEEE International Conference on Computer Vision*, 1999, pp. 116-122, vol. 1, doi: 10.1109/ICCV.1999.791206.

- [9] A. S. Konwar, B. S. Borah, and C. T. Tuithung, "An American sign language detection system using HSV color model and edge detection," *2014 International Conference on Communication and Signal Processing*, 2014, pp. 743-747, doi: 10.1109/ICCSP.2014.6949942.
- [10] S. Upendran, and A. Thamizharasi, "American sign language interpreter system for deaf and dumb individuals," *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, 2014, pp. 1477-1481, doi: 10.1109/ICCICCT.2014.6993193.
- [11] V. Bheda, and D. Radpour, "Using deep convolutional networks for gesture recognition in American sign language," *ArXiv, abs/1710.06836*, 2017.
- [12] S. Srinath, and G. K. Sharma, "Classification approach for sign language recognition," *2017 Int. Conf. on Signal, Image Processing Communication & Automation (ICSIPCA)*, 2017.
- [13] J. R. Pansare, and M. Ingle, "Vision-based approach for American sign language recognition using edge orientation histogram," *2016 International Conference on Image, Vision and Computing (ICIVC)*, 2016, pp. 86-90, doi: 10.1109/ICIVC.2016.7571278.
- [14] J. Singha, and K. Das, "Recognition of Indian sign language in live video," *International Journal of Computer Applications*, vol. 70, 2013, doi: 10.5120/12174-7306.
- [15] G. A. Rao, K. Syamala, P. V. V. Kishore, and A. S. C. S. Sastry, "Deep convolutional neural networks for sign language recognition," *2018 Conference on Signal Processing and Communication Engineering Systems (SPACES)*, 2018, pp. 194-197, doi: 10.1109/SPACES.2018.8316344.
- [16] D. Rathi, "Optimization of transfer learning for sign language recognition targeting mobile platform," *ArXiv, abs/1805.06618*, 2018.
- [17] <https://www.kaggle.com/datamunge/sign-language-mnist>
- [18] H. Zhao, Z. Xu, G. Han, and D. Xie, "The fast image recognition based on grayscale features," *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, 2012, pp. 730-734, doi: 10.1109/CSAE.2012.6273052.
- [19] A. Mikołajczyk, and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, 2018, pp. 117-122, doi: 10.1109/IIPhDW.2018.8388338.
- [20] R. Chauhan, K. K. Ghanshala, and R. C. Joshi, "Convolutional neural network (CNN) for image detection and recognition," *2018 1st International Conference on Secure Cyber Computing and Communication (ICSCCC)*, 2018, pp. 278-282, doi: 10.1109/ICSCCC.2018.8703316.
- [21] A. F. Agarap, "Deep learning using rectified linear units (ReLU)," *ArXiv, abs/1803.08375*, 2018.
- [22] J. Nagi et al., "Max-pooling convolutional neural networks for vision-based hand gesture recognition," *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, 2011, pp. 342-347, doi: 10.1109/ICSIPA.2011.6144164.
- [23] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929-1958, 2014.
- [24] X. Xu, S. Huang, L. Zheng, and L. Zhang, "The geometric structure of generalized softmax learning," *2018 IEEE Information Theory Workshop (ITW)*, 2018, pp. 1-5, doi: 10.1109/ITW.2018.8613303.
- [25] S. Mannor, D. Peleg, and R. Y. Rubinstein, "The cross entropy method for classification," *Proceedings of the 22nd International Conference on Machine Learning (ICML'05)*, 2005, pp. 561-568, doi: 10.1145/1102351.1102422.
- [26] G. Cong, O. Bhardwaj, and M. Feng, "An efficient, distributed stochastic gradient descent algorithm for deep-learning applications," *2017 46th International Conference on Parallel Processing (ICPP)*, 2017, pp. 11-20, doi: 10.1109/ICPP.2017.10.
- [27] D. P. Kingma, and J. Ba, "Adam: A method for stochastic optimization," *CoRR, abs/1412.6980*, 2014.
- [28] Z. Li, L. Jin, C. Yang, and Z. Zhong, "Hyperparameter search for deep convolutional neural network using effect factors," *2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, 2015, pp. 782-786, doi: 10.1109/ChinaSIP.2015.7230511.
- [29] <https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd>