EasyChair Preprint

№ 8969

Sign Language Recognition for Bangla Alphabets Using Deep Learning Methods

Md.Saiful Islam, Dhrubajyoti Das, Saurav Das and Md.Nahid Ullah

October 3, 2022

2022 4th International Conference on Sustainable Technologies
for Industry 4.0 (STI), 17-18 December, Dhaka

Sign Language Recognition for Bangla Alphabets Using Deep Learning Methods

No Author Given
No Institute Given

***Abstract*—Language is an essential aspect of communication. We can understand and communicate each other's feelings through language. However, certain members of our society cannot talk or usually listen, leaving them with only sign language as a means of communication. Although researchers put a lot of time and effort into deciphering sign languages, most of their efforts have been focused on sign digits, and some are limited to simple samples. To address these prevalent concerns in earlier research, we created a new dataset of Bangla alphabets consisting of 2340 samples with different backgrounds. We also proposed a custom CNN architecture and compared its performance with other state-of-the-art models like ResNet, EfficientNet InceptionV3, and VGG19. All state-of-the-art models were trained and evaluated with custom dataset weights and Imagenet weights, and the best results were compared to our custom CNN. Our custom CNN did better than all the state-of-the-art models on our dataset with 92% accuracy.**

***Index Terms*—Bangla sign alphabets, deep learning methods, complex background, custom CNN, transfer learning**

I. Introduction

Hearing impairment is a form of physical impairment that restricts a person's capacity to operate normally in all aspects of his life. Since hearing is an essential component of our lives, individuals with this handicap may have difficulty participating in social or cultural activities. Because of this issue, they may have trouble obtaining work that is suitable for them. As a consequence of this, they are unable to contribute to the economy of a nation and are ignored by society. Over 5% of the world's population, as reported [1] by the World Health Organization (WHO), suffers from debilitating hearing loss that curtails their daily lives and means of subsistence. An estimated 360 million persons have a hearing impairment, 32 million children. There are more than 30 lakhs impaired people in Bangladesh[2], but still, there are few opportunities for people to learn sign language. It is also essential for ordinary people, enhancing communication between normal and deaf and hard of hearing people. As communication is necessary for all sorts of people to sustain themselves in society, deaf and hard of hearing people must use sign language as a means of communication. Sign language is a system of hand gestures used by the deaf and speech-impaired to communicate[3]. It involves signs for fingers, hands, arms, head, body, and facial expressions. There are 50 letters and 10 digits in Bangla language [4]. According to studies [5], persons with impairments are more likely to be criminally assaulted. Their position is made worse by the lack of legal protections. Since sign language is not a legal court language, our judges are often hesitant to believe deaf people when they testify or to call sign language interpreters experts.

It is tough for a normal person to identify sign languages if he has no prior knowledge of them. It is difficult for a normal individual to understand an impaired person's sentiments without an interpreter. With the advent of Artificial Intelligence, several attempts have been made to improve the communication between normal and deaf and hard-of-hearing individuals. Furthermore, the various backgrounds of photographs and image-taking settings exacerbate these problems. In recent years, breakthroughs in computer vision and deep learning have enabled researchers to discover the deep properties of several sign language recognition systems, regardless of the image background or capture environment. Many works exist on recognizing digit sign languages using convolutional neural network (CNN) [6], [7], [8]. Furthermore, several research works have discovered Bangla sign language utilizing cutting-edge CNN architectures such as VGG16. in[9]. The primary flaw in these studies is that it uses almost identical data for training and testing, resulting in high accuracy only for the suggested dataset. Most of the works are focused on digit recognition, but characters in Bangla Sign Language are hard to interpret because most of them are identical. To overcome the issues mentioned earlier, we have developed a dataset consisting of samples in the complex background, and we have also suggested a CNN architecture to compare its performance with several state-of-art architectures like ResNet[10], EfficientNet[11], InceptionV3[12] and VGG19[13].

The contributions of this research work are as follows:

- Create a new dataset of 2340 images with varying background.
- Develop a custom CNN model for classification.
- Compare our custom CNN model with other CNN architectures on our dataset.

II. Related Works

Several research-based works on recognizing sign languages have been done in recent years. Machine learning methodologies were used in a few studies, while deep learning approaches were used in the majority.

Muttaki Hasan et al. [14] have suggested a machine learning hand gesture recognition system that employs HOG for feature extraction and an SVM classifier for classification. They used the Bangla Sign Language Dictionary to gather 16 Bangla sign language gestures and preprocessed the photos to 200X200 pixels. They achieved an accuracy of 94%.

In [15], Omkar Vedak et al. have also proposed a machine learning system approach to recognise the sign language images collected from videos broken into frame by frame. Their dataset consists of 6000 samples. They have used HOG as a feature detector. The recognition is done by template matching, and it produces 88% accuracy.

Fahad Yasir et al. [16] have presented a scale-invariant feature transform approach to recognize Bangla Sign Language. They preprocessed the photos using Gaussian distribution and gray scaling algorithms. They also employed K-means clustering on descriptors that had already been calculated using the SIFT method. Finally, for each sign word, they utilized a binary SVM classifier using a separate dataset.

Touhidur Rahman et al. suggested a Convolution Neural Network-based numerical sign language identification system based on computer vision [17]. Their system uses a webcam to capture photos, which are then segmented for skin colour and converted to binary to extract features. If the same individual inputs the data, their system generates 90% accuracy.

In [18], Pias Paul presented two custom CNN-based model on 24 static ASL signs. They also compared their models to models that had been pre-trained. Their datasets were scaled to 200X200 pixels. They have also utilized augmentation techniques to boost the data's variety. Their ASL dataset's proprietary models produce 86.52% and 85.88%.

On 36 static bangla sign languages, Md. Sanzidul Islam et al. suggested a multilayered CNN approach [19]. Their photos have been reduced to 120X120 pixels. In their study, they employed ten convolution layers. They proposed a dataset with 50 samples for each class and 1800 samples. Within 30 epochs, they were able to obtain a 92% accuracy.

Thasin Abedin et al. [20] have proposed a new BDSL network that is concatenated. This architecture comprises an image network and a posture estimation network. They could recognize 38 distinct indicators, with a total of 11061 samples in their collection. Their method yields a 91.51% accuracy rating.

In [21] Md. Shahinur Alam et al. created a convolutional neural network architecture that recognizes and converts the Bangla Sign Language into textual Bangla characters. They utilized a 4600-sample dataset. Using a high-definition webcam, they gathered data from volunteers. Their technique yields an accuracy of 99.57% on validation data.

Ankita Wadhaman et al. [22] suggested a CNN model for recognizing 100 Indian Sign Language signs from static photos. They employed a combination of optimizers and CNN models to select the optimum model. A total of 35,000 photos were collected from 100 users for their dataset. Their CNN has a 99.90% accuracy rate.

For identifying sign language, Md. Mehdi Hasan et al. [23] presented a unique CNN model. They utilized the Esharalipi dataset, which is quite popular. They claim that their approach outperforms all prior studies using the Esharalipi dataset. Their model has a 99.2% accuracy rate.

In [8], MD Shafiqul offered an extensive dataset of 30916 samples of Bangla Sign Language. They also suggested a CNN model and examined its effectiveness in separating alphabets and letters and merging both. They stated that their model yields an accuracy of 99.83%, 100%, and 99.80% on characters, numbers, and a combination of the two.

To reduce class similarities, Kanchon Kanti Podder et al. designed a colour-coded fingertip pattern [24]. They used ResNet18 to create a transfer learning technique. They have also put up a large dataset with 45,958 samples. They claim to have reached an accuracy of 99% by using a transfer learning technique on their dataset.

Farhad Yasir et al.[25] presented a learning-based convolution neural network approach. Throughout the entirety of their approach, they make use of a Leap Movements Controller in order to track the movements of their hands continuously. The Leap Motion controller supplies information on the hand's position, orientation, rotation, and other non-linear features. They stated that using their basic sign expressions; they could attain a 3% error rate.

III. Method and Materials

This section explains the approach we suggest for identifying Bangla Sign Language. The approach begins with constructing a new training dataset, followed by a description of standard datasets used in our study, the development of a custom CNN model, deep feature extraction for training the model, and finally, Sign Language classification for Bangla Alphabets.

A. Architecture of the proposed model

For the objective of finding sign languages for the Bangla alphabet, we proposed a CNN model. The structure of the model consists of an input layer, Conv1, Pooling1, Conv2, Pooling2, Conv3, Pooling3, Conv4, Pooling4, two fully connected layers, and a softmax layer as shown in Fig. 1 for an input image of size $w \times h$.

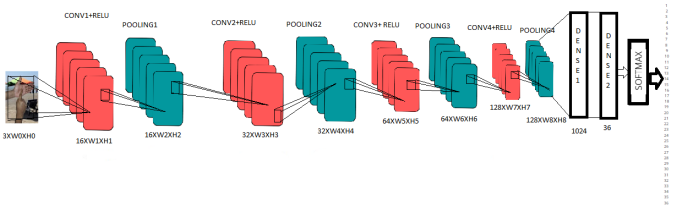

Fig. 1: Architecture of our proposed model

B. Training and Classification of Alphabets Sign Characters

In our custom sign language recognition model, our approach extracts in-depth characteristics of sign characters. Activation in each layer of the model converts the specific information included in the picture of the input sign character into a more abstract representation and summarizes the character's primary characteristics as the image moves deeper into the model's structure. The softmax layer is then used to categorize more profound and more precise data as a feature to improve the classification's accuracy.

Our model is initially fed images in batches so that it may learn and adjust its parameters in convolution, pooling, and fully connected layers of the network, resulting in a 1×1024 vector that summarizes the features. These characteristics are transmitted through a second thick layer, resulting in a 1×36 vector. After that, this vector is sent to the softmax layer, which uses it to categorize sign alphabets into the appropriate category. Before verifying the model and its associated parameters using a collection of validation images, we transit the training images through a series of epochs. Our model's loss function is categorical cross-entropy.

C. Transfer Learning Models

Our CNN model was compared to the state-of-the-art architectures described in Table I using a variety of Transfer Learning model approaches.

TABLE I: Transfer learning techniques applied for comparison

Model	Version
EfficientNet	B0
Inception	V3
VGG	19
RestNet	50

IV. Result and Observation

This section describes the results of our model's experimentation and analysis. The study was carried out on a 2vCPU @ 2.2GHz. With 13GB RAM. and 100GB Free Space n1-highmem-2 instance. The suggested models are written in Python using the Keras and TensorFlow libraries.

A. Dataset

We created a novel dataset with 2340 samples. To ensure that our dataset includes real-life events, we evaluated all circumstances, such as shadow factors, poor light situations, complicated background objects, and regions of interest conflicting with the object of interest. We created individual scenarios for each sample in our dataset so that our model may learn real-world circumstances and perform well in them. To ensure that the samples were diverse, all photographs were captured using different phone cameras. A total of 2399 RGB-colored 128x128 sized images representing 36 Bangla Sign Language symbols are analyzed. 80% of the data was utilized for training, and the remaining 20% was used for testing. Details are provided in Table II.

TABLE II: Training and test samples of each class

Class	Training Samples	Test Samples	Class	Training Samples	Test Samples
ং	52	13	ঠ	52	13
ঃ	52	13	ড	52	13
অ	52	13	ঢ়	52	13
আ	52	13	ঢ	52	13
ই	52	13	ত	52	13
উ	52	13	থ	52	13
এ	52	13	ধ	52	13
ও	52	13	ন	52	13
ক	52	13	প	52	13
চ	52	13	ফ	52	13
গ	52	13	ব	52	13
ঘ	52	13	ভ	52	13
ল	52	13	ম	52	13
ছ	52	13	য়	52	13
জ	52	13	র	52	13
ঝ	52	13	খ	52	13
ট	52	13	স	52	13
দ	52	13	হ	52	13

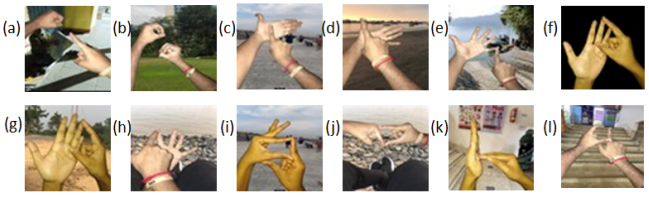

Fig. 2: Samples of our dataset: (a) ং , **(b)** ঃ , **(c)** অ , **(d)** আ, **(e)** ই, **(f)** উ, **(g)** এ, **(h)** ও, **(i)** ক, **(j)** চ, **(k)** গ, **and (l)**ঘ

a) Adding Shadow Factor: We have added shadow factors in every class to ensure that our model recognises the sign characters well, considering the shadow factor included in the images. Fig. 3 demonstrates some of the samples in which shadow factors were included.

Fig. 3: Example of samples included with shadow factors

b) Adding Complex Background: Most models are based on samples with simple backgrounds; to address these concerns, we created a new dataset of examples with complicated backgrounds, as shown in Fig. 4.

Fig. 4: Example of samples included with shadow factors

B. Augmentation Techniques

An example of a data augmentation strategy involves adding slightly changed copies of current data or synthesizing new data from existing data to the data set. In the process of training a deep learning model, it acts as a regularizer and helps to prevent overfitting. To enhance the diversity of data, we have used some of the techniques mentioned in Table III.

TABLE III: Data augmentation techniques

Augmentation Techniques	**Parameters**
Horizantal Flip	True
Vertical Flip	True
Rescale	1./255
Zoom_Range	0.3
Shear_Range	0.2

C. Hyper-parameters of our Model

When developing our model, we employ categorical cross-entropy as our loss function. A maximum of 200 iterations of our model were performed, with an early stop set to a patience value of 15. The loss function is optimized using Adam, a loss function optimizer. The hyper-parameters given the most careful attention in our model are detailed in Table IV. In our trials, the default settings for the Conv1, Conv2, Conv3, and Conv4 layers employ 16, 32, 64, and 128 5×5 filters, while the pooling layers in our model use 2×2 max pooling.

TABLE IV: Hyperparameters used in our custom model

Hyperparameters	**Value(s)**
Batch size	32, 18
Optimizer	Adam
Epochs	200
Loss function	Categorical cross-entropy

a) Batch Size: The size of a batch has a significant effect on how a model learns. Batch sizes 32 and 18 are used in our CNN model. Using our model, we found that a batch size of 32 works best, with the rest of the parameters remaining the same.

b) Optimizer: Several Optimizers were applied to our custom model to select the optimum Optimizer. We found out that adam works best for our CNN model, as shown in Table V.

TABLE V: Accuracy of different optimizers on our custom model

Optimizers	**Accuracy**
Adam	92%
RMSProp	90%
SGD	64%

c) Activation Functions: Experimenting with many activation functions, such as ReLU, tanh, and sigmoid, in each convolution layer of our custom model for sign language recognition allows us to improve the accuracy of our results. As shown in Table 7, the ReLU activation function yielded the best test accuracy for our model when run with the default settings.

TABLE VI: Accuracy of different activation functions on our custom model

Activations	**Accuracy**
Relu	92%
Tanh	86%
Sigmoid	4%

d) Dropout Rates: In our model, we explored a few dropout rates ranging from 10% to 35%. From our experiments, we concluded that 30% dropout yields the best accuracy, which is evident in Table VII.

TABLE VII: Accuracy of different dropout rates on our custom model

Dropout Rates	0.10	0.15	0.20	0.25	0.30	0.35
Accuracy	0.85	0.89	0.89	0.90	0.92	0.78

e) Early Stopping: We employed early stopping to cease training if the validation accuracy did not improve in 15 consecutive epochs to avoid confusion about the optimal epoch.

D. Performance Analysis

We created a novel dataset of 36 Bangla signs having 2340 samples and also checked the generality of our proposed model. Fig. 5 shows the comparison based on accuracy between different models. Our custom CNN model's accuracy and loss curves are given in Fig. 6 and Fig. 7. We have also tested our custom CNN model with two different datasets[8], [19], which is shown in Table VIII.

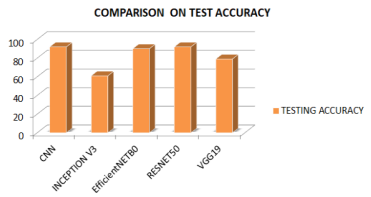

Fig. 5: Accuracy of different models on our dataset

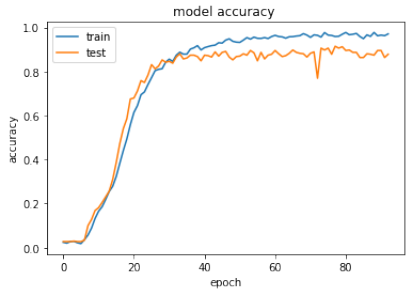

Fig. 6: Accuracy Curve of our custom CNN model

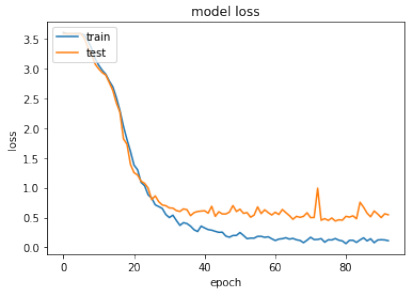

Fig. 7: Loss Curve of our custom CNN model

TABLE VIII: Classification results on other datasets

Dataset	Classes	Accuracy
BDSL[8]	38	99%
Esharalipi[19]	36	87%

V. Conclusion

This paper presents a dataset with 2340 samples and proposes a custom CNN model classification system for Bangla sign languages. Our model has been trained to detect Bangla sign language in shadow factors, poor light situations, complicated background objects, and regions of interest. We have used two benchmark datasets for analyzing our model. Our proposed CNN achieved 87% on isharalipi, 99% on BDSL dataset and 92% accuracy on our proposed dataset. Furthermore, because our model has fewer network parameters, it is effective for smaller devices. Even though our model is accurate, we enhanced its dependability and resilience by testing it on diverse data. We focused on categorizing sign language images with complicated backdrops and varying lighting conditions. In future, we will enhance our dataset by including more users and classes. We will also use the transformer-based model to find out how well our dataset works.

References

[1] "Who global estimates on prevalence of hearing loss," 2014 (accessed 8 october, 2021). [Online]. Available: https://www.who.int/pbd/deafness/WHO_GE_HL.pdf

[2] K. H. Tarafder, N. Akhtar, M. M. Zaman, M. A. Rasel, M. R. Bhuiyan, and P. G. Datta, "Disabling hearing impairment in the bangladeshi population," *The Journal of Laryngology amp; Otology*, vol. 129, no. 2, p. 126–135, 2015.

[3] A. S. M. Miah, J. Shin, M. A. M. Hasan, and M. A. Rahim, "Bensignnet: Bengali sign language alphabet recognition using concatenated segmentation and convolutional neural network," *Applied Sciences*, vol. 12, no. 8, p. 3933, 2022.

[4] "Importance of sign language," 2021 (accessed 8 october, 2021). [Online]. Available: https://www.thedailystar.net/the-silent-conversation-37929

[5] "Problems of dumb and deaf people," 2021 (accessed 8 october, 2021). [Online]. Available: https://www.ncjrs.gov/ovc_archives/factsheets/disable.htm

[6] M. S. Islam, S. S. S. Mousumi, N. A. Jessan, A. S. A. Rabby, and S. A. Hossain, "Ishara-lipi: The first complete multipurposeopen access dataset of isolated characters for bangla sign language," in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*. IEEE, 2018, pp. 1–4.

[7] S. R. Kalbhor and A. M. Deshpande, "Digit recognition using machine learning and convolutional neural network," in *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, 2018, pp. 604–609.

[8] M. S. Islalm, M. M. Rahman, M. H. Rahman, M. Arifuzzaman, R. Sassi, and M. Aktaruzzaman, "Recognition bangla sign language using convolutional neural network," in *2019 international conference on innovation and intelligence for informatics, computing, and technologies (3ICT)*. IEEE, 2019, pp. 1–6.

[9] M. Hossen, A. Govindaiah, S. Sultana, and A. Bhuiyan, "Bengali sign language recognition using deep convolutional neural network," in *2018 joint 7th international conference on informatics, electronics & vision (iciev) and 2018 2nd international conference on imaging, vision & pattern recognition (icIVPR)*. IEEE, 2018, pp. 369–373.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[11] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] M. Hasan, T. H. Sajib, and M. Dey, "A machine learning based approach for the detection and recognition of bangla sign language," in *2016 International Conference on Medical Engineering, Health Informatics and Technology (MediTec)*. IEEE, 2016, pp. 1–5.

[15] O. Vedak, P. Zavre, A. Todkar, and M. Patil, "Sign language interpreter using image processing and machine learning," *International Research Journal of Engineering and Technology (IRJET)*, 2019.

[16] F. Yasir, P. C. Prasad, A. Alsadoon, and A. Elchouemi, "Sift based approach on bangla sign language recognition," in *2015 IEEE 8th international workshop on computational intelligence and applications (IWCIA)*. IEEE, 2015, pp. 35–39.

[17] M. T. Rahman, M. M. Morshed, M. Hasanuzzaman, and M. I. Jabiullah, "A computer vision-based real time bangla numerical sign language recognition using convolutional neural networks (cnns)," *ResearchGate. https://www. researchgate. net/publication/335292263 (Accessed Apr. 18, 2021)*.

[18] P. Paul, M. Bhuiya, M. Ullah, M. N. Saqib, N. Mohammed, S. Momen *et al.*, "A modern approach for sign language interpretation using convolutional neural network," in *Pacific Rim International Conference on Artificial Intelligence*. Springer, 2019, pp. 431–444.

[19] M. Islam, S. S. S. Mousumi, A. Rabby, S. A. Hossain *et al.*, "A simple and mighty arrowhead detection technique of bangla sign language characters with cnn," in *International Conference on Recent Trends in Image Processing and Pattern Recognition*. Springer, 2018, pp. 429–437.

[20] T. Abedin, K. S. Prottoy, A. Moshruba, and S. B. Hakim, "Bangla sign language recognition using concatenated bdsl network," *arXiv preprint arXiv:2107.11818*, 2021.

[21] M. Alam, M. Tanvir, D. K. Saha, S. K. Das *et al.*, "Two dimensional convolutional neural network approach for real-time bangla sign language characters recognition and translation," *SN Computer Science*, vol. 2, no. 5, pp. 1–13, 2021.

[22] A. Wadhawan and P. Kumar, "Deep learning-based sign language recognition system for static signs," *Neural computing and applications*, vol. 32, no. 12, pp. 7957–7968, 2020.

[23] M. M. Hasan, A. Y. Srizon, and M. A. M. Hasan, "Classification of bengali sign language characters by applying a novel deep convolutional neural network," in *2020 IEEE Region 10 Symposium (TENSYMP)*. IEEE, 2020, pp. 1303–1306.

[24] K. K. Podder, M. Chowdhury, Z. B. Mahbub, and M. Kadir, "Bangla sign language alphabet recognition using transfer learning based convolutional neural network," *Bangladesh J. Sci. Res*, pp. 31–33, 2020.

[25] F. Yasir, P. Prasad, A. Alsadoon, A. Elchouemi, and S. Sreedharan, "Bangla sign language recognition using convolutional neural network," in *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*. IEEE, 2017, pp. 49–53.