# Optimization of Deep CNN-based Bangla Sign Language Recognition using XGBoost classifier

Raima Islam*
raima.islam@g.bracu.ac.bd
Deptartment of Computer Science and Engineering (CSE)
School of Data and Sciences (SDS)
BRAC University
Dhaka, Bangladesh

Mashequr Rahman Khan*
mashequr.rahman.khan@g.bracu.ac.bd
Deptartment of Computer Science and Engineering (CSE)
School of Data and Sciences (SDS)
BRAC University
Dhaka, Bangladesh

Md Zunayedul Islam*
md.zunayedul.islam@g.bracu.ac.bd
Deptartment of Computer Science and Engineering (CSE)
School of Data and Sciences (SDS)
BRAC University
Dhaka, Bangladesh

MD. Mustakin Alam†
md.mustakin.alam@g.bracu.ac.bd
Deptartment of Computer Science and Engineering (CSE)
School of Data and Sciences (SDS)
BRAC University
Dhaka, Bangladesh

Md Sabbir Hossain†
md.sabbir.hossain1@g.bracu.ac.bd
Deptartment of Computer Science and Engineering (CSE)
School of Data and Sciences (SDS)
BRAC University
Dhaka, Bangladesh

Annajiat Alim Rasel‡
annajiat@gmail.com
Deptartment of Computer Science and Engineering (CSE)
School of Data and Sciences (SDS)
BRAC University
Dhaka, Bangladesh

## ABSTRACT

Deafness is one of the prime health problems in Bangladesh where almost 10% of the population fall under that category. The plight of the deaf and dumb to fit in with the mass population is quite big. Unable to express themselves through sound, they have to rely on sign language in order to communicate. Through recent studies, it was found that people with mutism are often subjected to heavy discrimination and exclusion from social settings. In order to create a more inclusive environment for the minority, we developed a model that will recognize Bengali characters and digits expressed using sign language which corresponds to the BdSL guidelines. The model is made up of two parts: a convolutional neural network that uses deep learning techniques, and an Extreme Gradient Boosting(XGBoost) classifier. Together, these components form the full architecture. The execution of the model has been validated using the 'Ishara-Lipi' dataset which is the first open-access digit and character dataset for Bangla Sign Language (BdSL). With the help of pre-processing techniques such as contrast limited adaptive histogram equalization, using a more complex pre-trained CNN model like Inception-ResNet-v2, and optimizing the XGBoost model by using GridSearchCV, we were able to achieve an accuracy of 86.67% in the character recognition and an accuracy of 97.33% in digit recognition. This paper contains the extensive literature on the model created along with existing work attempted for sign language detection.

## CCS CONCEPTS

• **Applied computing** → **Computer-aided design**; • **Computing methodologies** → **Cross-validation**.

## KEYWORDS

model, deep convolutional neural network, XGBoost, Bangla sign language, work, accuracy, order, population, dataset, recognition

**ACM Reference Format:**
Raima Islam, Mashequr Rahman Khan, Md Zunayedul Islam, MD. Mustakin Alam, Md Sabbir Hossain, and Annajiat Alim Rasel. 2023. Optimization of Deep CNN-based Bangla Sign Language Recognition using XGBoost classifier. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (ICCTA '23).* ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

## 1 INTRODUCTION

Sign Language is the primary method of communication between humans who are deaf, hard-hearing and those who suffer from mutism. This language uses hand shapes, palm orientation, movement, and body position to convey expressions through visual signs, rather than using sound. The goal is to create a language that can be conveyed visually, using the body to communicate [1]. Sign languages have developed into useful means of communication and are the foundation of local Deaf cultures wherever deaf societies exist.

With 265 million speakers, Bangla is the seventh most spoken language in the world, but there are only a few studies and possibilities for working on automatic Bengali sign language detection. Almost 14 million people in Bangladesh are hard of hearing or deaf. The majority of these people do not have the facilities for hearing aids, education or basic aspects of life such as employment [2].

*Authors of this research
†Co-supervisors of this research
‡Supervisor of this research


*ICCTA '23, May 10–12, 2023, Vienna, Austria*

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

Thus, in terms of socio-economic concepts, deaf and mute people are very much secluded despite having the desire to enter the workforce or having basic needs. In recent years, a lot of computer-aided models which recognise Bengali Sign Language have been developed which have performed very well in real time detection. Models like this one can help bridge the communication gap between people who can hear and those who are hearing-impaired or deaf. By providing a visual means of communication, these models can make it easier for people with hearing loss to communicate with others [3].

One of the biggest constraints in such sign language detection is the size of the datasets and access to open-source datasets. Even though the Ishara-Lipi dataset is considered the standard BdSL dataset, the amount of pictures is not enough to achieve high accuracy. Most of the previous work done on BdSL recognition (Haque et al., 2019, Rahman et al., 2012, Uddin et al., 2017) utilized a limited amount of data, thus increasing the unreliability of sign recognition. We aimed to create a model that will not only make a social contribution to the minorities in need but also overcome the challenges of small datasets by using various image and algorithm optimization techniques.

We created a Deep Convolutional Neural network as DCNNs has proved to be beneficial for any pattern detection and computer vision task [4–6]. With a total of 7 layers, we optimized the model by adding batch normalization which resulted in speeding up the training set, max pooling which performed model averaging along with the dropout layer, which resulted in reduced overfitting. To make the model more robust, we have employed Inception-ResNet-v2 which incorporates residual connection and outperformed its predecessors such as ResNet-50 and Inception-v3. To improve the Extreme Gradient Boosting algorithm, we used the Grid search method for tuning the hyperparameters which resulted in an increase of the accuracy in the field of Bangla Sign Language Detection. Therefore, we have created a hybrid model that utilizes a Deep-CNN model with pre-trained weights and this model is then fed into the XGBoost algorithm, creating a deep learning-based classifier that effectively recognizes sign languages. Our proposed architecture has been tested against other existing models, and it outperformed some of those.

## 2 LITERATURE REVIEW

Huang's paper presented a Sign Language detection system that included various methods regarding gesture detection, tracking of hands which is model-based, and extraction of features[7]. He measured Hausdorff distance to track variations in shapes, Fourier descriptor for characterizing features, and 3D modified Hopfield Neural Network (HNN) for gesture recognition. The author tested their model on 15 different moving hand gestures and achieved the result of 91% accuracy. This paper takes sign language detection to another level by training on slightly moving gestures. The system described by the author deals with simple hand gestures since processing images frame by frame is pretty difficult. However, this process seems promising for precise training of simple Bangla Sign Language detection.

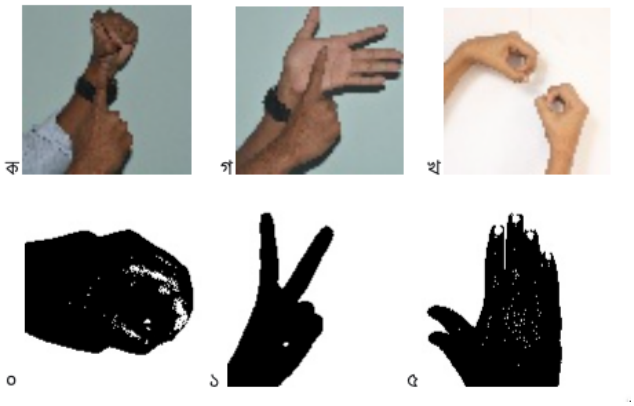


**Figure 1: Some symbolic representation in BSL**

Yasir and Khan's research focuses on communication with Bangladeshi sign language users with the aid of computer vision [8]. It incorporates skin detection, preprocessing, multiple machine learning methods like Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA), and to train-test the model, neural network was employed. The authors implement two-handed gesture recognition with LDA and Artificial Neural Networking (ANN). The process of the system has 4 phases: Skin Extraction, Feature Extraction, Training, and Recognition. The dataset consisted of 15 Bangla letters from 22 individuals, totalling in 330 RGB images. The RBG images underwent skin detection aided by the YCbCr Algorithm, and after that, they undergo 2 bit grayscale conversion, and then PCA, LDA, and ANN are applied to extract the features. Success rate of PCA is 26% whilst the success rate of LDA is at 100%, the poor state of the PCA-NN could be tied to over-fitting.

The study presented a computational approach which is geometrical SIFT-based to actively and efficiently perceive Bengali sign language [9]. Using scale invariant feature transform (SIFT), the detection of distinct attributes of a picture is performed. It is observed that SIFT is actually better suited for such an approach. For the paper, the authors worked with two handed gestures which have been captured from continuous static im. Bag of Words model is incorporated as well to establish a visual vocabulary dictionary for BdSL. K-means clustering algorithm is executed to cluster homogeneous ones on all the training descriptors into a single bag. For the training dataset, the study made use of 9 signs of Expressions and 6 signs of alphabets, SVM classifier is implemented here. For sign of expression, K-NN's recognition rate ranges from 70% to 100%, whilst SVM's is 80% to 100%. For Bangla vowels, K-NN's range is 95% to 99% whilst SVM's is 80% to 100%.

Hasan et al's paper focuses on providing speech impaired individuals a voice to better integrate them with mainstream society [10]. In the paper, hand gesture recognition is done with the help of Histogram of Oriented Gradients (HOG) for extracting image characteristics and Support Vector Machine (SVM) for the purpose

of classification. The entire process is divided into the six following steps: pre-processing an image, segmenting an image, extracting the features of the image, using a feature vector to train the classifier, recognition and prediction, and producing an output which is audio. The paper proposes a procedure for 16 gesture recognition which are predefined by applying HOG features. SVM is used for decision making given the large dataset. The signs have been collected from " বাংলা ইশারা ভাষার অভিদান ". The accuracy could be improved with an increased number of training images, whilst the gesture recognition rate could also be improved by better adjustment, orientation, and lighting of the sign images.

The research investigates the existing works that are implemented to develop Bangla Sign Language Translation systems, it also proposes its own solution which is a bidirectional Bangla Sign Language translator [11]. The intent behind reviewing the existing systems is to completely understand the necessary factors needed to build a translation system that would outperform all the previous works, the limitations of those works have been taken into consideration as well. In total, 22 research papers were collected for this paper, and out of those 22, 15 were relevant to Bangla Sign Language. The proposed system follows a process that is broken down into 4 steps: a text input is taken, processing the input based on Bangladeshi Sign Language (BdSL) requirements and grammar, looking up the processed text in the database, and finally displaying the output on a monitor which could be in Bangla Sign Language Text format, a video format, or an image format.

One of the most common issues of computer vision is Sign Language Recognition, which poses a lot of obstacles that need to be overcome. Just like natural languages, even sign languages change over time . As a result, it is necessary to develop scalable strategies for dealing with such variations and difficulties. To solve this problem, a spatio-temporal-based solution is proposed which pays attention to hand and full body parts with the aid of 3D-CNNs and LSTMs. It learns it in an end-to-end style for zero-shot sign language recognition (ZSSLR) where a new criterion dataset for ZSSLR known as ASL-Text is defined and the issue of ZSSLR is formulated. The overall proposed framework achieves 20.9% top-1 normalized accuracy and 51.4% top-5 normalized accuracy [12].

Kulhandjian et al's paper studied American Sign Language using Doppler radar and tried to detect it using their deep learning model [13]. Using a microwave X-band Doppler radar transceiver, they captured different ASL hand motions as micro-Doppler signals. The authors put those Doppler signatures in the spectrogram after applying time-frequency analysis. Then they used DCNN for initial training and VGG-16 for further deep learning of the model. Their final result was 95% accuracy which is a fantastic job for ASL detection. This method is slightly more expensive than others but worth trying for Bangla Sign Language detection.

Hasan et al's paper proposed a novel Convolutional Neural Network to detect Bangla Sign Language [14]. Their model was implemented with Keras on the top of TensorFlow and trained over 36 Bengali alphabets that were incorporated into 1800 alphabets in the final dataset. They developed their CNN with 6 convolutional layers and 3 ReLU activation functions. Since they dealt with 36 Bengali characters, they used 24 softmax functions at the output layers of their architecture. The authors first preprocessed the data by converting RGB images into grayscale images. Then they applied different changes into each picture of each class (alphabets) and gathered 3600 images. They trained their model for 200 epochs until a constant learning rate. Finally they achieved an outstanding result of 99.22% accuracy. However, this model seemed to overfit the dataset that small.

This paper investigated the current state-of-the-art Bangla Sign Language detection and necessity of its generalization in order to deploy deep learning models for diversified datasets [15]. The authors conducted experiments on different inter-datasets and intra-datasets and found the result of intra-datasets very promising. They basically tried to explain why the efficiency of this study varies from researcher to researcher. That's why they investigated the virtualization of GradCAM and trained on Ishara-Lipi Dataset and BdSL Dataset. They used different architectures like AlexNet, ResNet, VGG, etc. After training the models, the authors evaluated them on different angular loss functions. The results discovered that optimal execution from VGG-19 model incorporated with SphereFace loss function achieved 55.93% and 47.81% on the inter-dataset. However, this paper fails to provide any distinctive stance on developing BSL detection.

## 3 DATASET

We collected images of Ishara-Lipi [16] BdSL Dataset from Kaggle. These images contain Bangla sign language digits and characters gathered from many Bangladeshi institutions based in Dhaka. There are 1075 images of BdSL digits and 1005 images of BdSL characters. Characters are in two classes - vowels and consonants. In this dataset, there are 6 classes of vowels and 30 classes of consonants with 50 images per class.

## 4 RESEARCH METHODOLOGY

In this research work, we created a Deep-CNN-based architecture combined with Inception-ResNet-v2 for classifying the training data for both characters and digits and then optimizing the accuracy by merging the DCNN with XGBoost. It contains five deep convolutional layers with two Dense (fully connected) layers, therefore making it a total of seven learned layers.

For preprocessing, we used CLAHE to improve the sharpness and visibility of the images. Then we applied the elimination of background [3] to the images to reduce any unwanted noise or objects. Following that, the images were divided by the highest grey level (255) to normalize them [17]. This means all of the pixel values were scaled to be between 0 and 1. The images were then resized to 75x75 to prepare the images for analysis and to make them more suitable to use with the model.

Before making the DCNN architecture, we created an Inception-ResNet-v2 model that takes an input shape of 75x75x3 and the activation classifier employed here is Softmax to improve the reliability. Then we made the DCNN model combined with the Inception-ResNet-v2 model which contains five Conv2D layers with (5,5) kernel size along with MaxPooling layers with (3,3) size which results in lesser chances of overfitting. This time, for the activation function, we chose Rectified Linear Unit (ReLU) because its gradient is not saturated, significantly speeding up stochastic gradient descent's convergence. To prevent further model overfitting, we have added a 20% dropout rate between both the Dense layers. At last, we add the 36 classes for the character dataset and 10 classes for the digit dataset to the DCNN architecture.

We then created the Extreme Gradient Boosting model (XGBoost) which is Friedman et al.'s (2001) effective and scalable implementation of the gradient boosting framework [18]. We extract the features from the DCNN after one-hot encoding of the labels [3]. Those features are then reshaped and fed to the XGBoost model . In order to finetune the hyperparameters of the XGBoost architecture, we employed GridSearchCV. We chose parameters of a maximum features array of [2,4,6], maximum depth of [80, 90, 100, 110] and estimators of [100, 200, 300, 1000]. The XGBoost model was fed into the grid search method and the accuracy was acquired in this way.

## 4.1 Data Preprocessing

*4.1.1 Background Elimination.* The images undergo the elimination of background to have any redundant and unnecessary features which may interfere with the classification process removed. How it works is that the background is distinguished from the object of focus and then is replaced with only the picture of the object with black pixels. Copies of the grayscale version of the images are made to which thresholding is applied to - the threshold value is contrasted with the pixel value, after which the pixel value is set according to the threshold value. Here, the implementation of the Gaussian adaptive threshold system was done which creates an output of a binary image in which black pixels resemble the background and white pixels resemble the foreground [19].

The following equation (Eq. (1)) used to compute the pixel's threshold number as:

$$T(i, j) = GMean(i, j) - C \tag{1}$$

*T(i,j)* is the pixel's threshold value located at the coordinates *(i,j)*. *GMean(i,j)* is the Guassian mean and is an arithmetic mean in which pixel values which are distant from *(i,j)* are emphasized weights of reduced amount. Constant $C$ is a value used to tune the threshold.

The following equation (Eq. (2)) is utilized to compute the 2D Gaussian distribution:

$$G(x, y) = \frac{1}{2\pi\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2}$$

$\sigma$ is denoted as the standard deviation of the distribution and is zero is inferred to be its mean value.

The following equation (Eq. (3)) used to modify the image to binary from grayscale:

$$I_{dst} = \begin{cases} 0, & \text{if } I_{src}(x, y) > T(x, y) \\ 255, & \text{otherwise} \end{cases} \tag{3}$$

$I_{src}$, $I_{dst}$ represent the magnitude values at *(x,y)* coordinates for the source and destination image respectively. *T(x,y)* is every individual pixel's threshold

In order to remove the noise, morphological closing transformations and morphological opening are applied on the binary after the threshold has been applied. Two morphological operations - erosion and dilation - are carried out in order to create a morphological opening of image A using B.

$$\textit{Erosion Operation} A \ominus B = \{z|(B)_z \subseteq A\} \tag{4}$$

$$\textit{Dilation Operation} A \oplus B = \{z|[\hat{B}_z \cap A] \subseteq A\} \tag{5}$$

$A$: foreground pixels
$B$: structuring element
$z$ is foreground value. [19]

*4.1.2 Data scaling.* In our proposed system, for optimization, the method applied is stochastic gradient descent. A large range of feature values creates erratic adjustments to the gradient descent algorithm, so to guarantee that the update process is very consistent, the feature value undergoes normalization - from 0 to 1. To do this, all the pixels' values were multiplied by 1/255. In doing so, the model is able to approach close to the point of minimum value much quicker due to fewer steps.

*4.1.3 Contrast Equalization.* Contrast Limited Adaptive Histogram Equalization (CLAHE) is a way for improving the contrast of images. It is based on adaptive histogram equalization, but it has the added benefit of limiting the amplification of contrast to reduce the amplification of noise. This makes the resulting images clearer and more usable. The slope of the transformation function in CLAHE determines how much contrast is amplified around a particular pixel value. CLAHE works by calculating the slope of the cumulative distribution function (CDF) for the neighbourhood around each pixel in the image. This slope is proportional to the histogram value at that pixel, and it determines how much the pixel's value will be changed. By clipping the histogram at a predetermined value before calculating the CDF, CLAHE prevents the amplification of contrast and reduces noise amplification. This constraint on the slope of the CDF limits the transformation of the image and improves its quality.

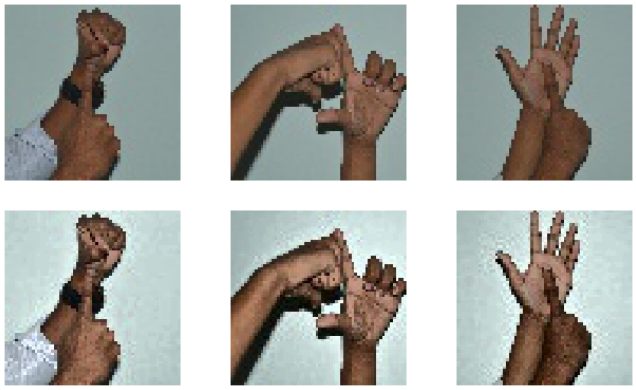

**Figure 2: Contrast Equalization**

## 4.2 Model Description

*4.2.1 Deep Convolutional Neural Networks.* The number of hidden layers in a deep convolutional neural network (DCNN) is what sets it apart from the conventional Convolutional Neural Network (CNN) in the field of deep learning. DCNN is made up of a lot of layers of neural networks, usually more than five hidden layers, with the goal of extracting more features and improving prediction accuracy. Two distinct types of layers—convolutional and pooling—are typically alternated in DCNN. The depth of each filter grows as the network moves from left to right. The last stage is typically made of at least one completely associated layers.

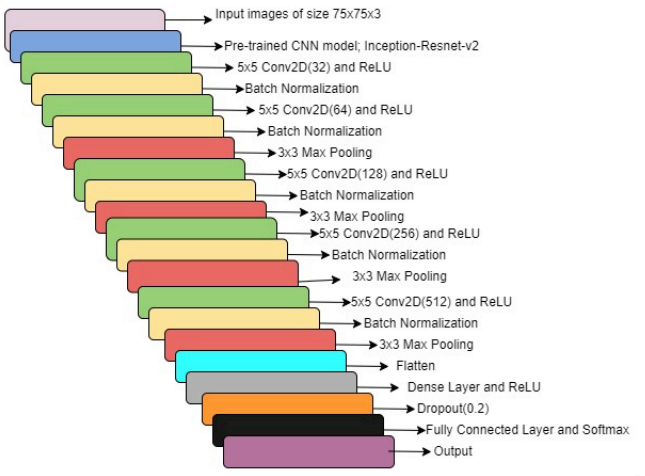


**Figure 3: The proposed DCNN architecture**

*4.2.2 eXtreme Gradient Boosting.* XGBoost (eXtreme Gradient Boosting) is a machine learning system for tree boosting that can handle large amounts of data and is available as open source software. One of the key features of the XGboost is its scalability in all scenarios which contributes to its success. XGBoost is a powerful machine learning algorithm that has several key features that make it effective. Some of these features include its ability to penalize trees, shrink leaf nodes proportionally, use Newton Boosting, and implement automatic feature selection. It can operate on a single computer or on a network of computers, and it can handle large amounts of data that do not fit in memory by using disk storage, which allows it to handle large datasets efficiently. These features make XGBoost a versatile and effective tool for many different types of machine learning tasks. In function space, XGBoost uses a Newton-Raphson method. XGBoost algorithm:

Input: training set $\{(x_i, y_i)\}_{i=1}^{N}$, a differentiable loss function $L(y, F(x))$, a number of weak learners $M$ and a learning rate $\alpha$.

Algorithm:

1. Initialize model with a constant value:

$$\hat{f}_{(0)}(x) = arg_{\theta} min \sum_{i=1}^{N} L(y_i, \theta) \tag{6}$$

2. For $m = 1$ to $M$:

i) Compute the 'gradients' and 'hessians':

$$\hat{g}_m(x_i) = [\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}]_f(x) = \hat{f}_(m-1)(x) \tag{7}$$

$$\hat{h}_m(x_i) = [\frac{\partial^2 L(y_i, f(x_i))}{\partial f(x_i)^2}]_f(x) = \hat{f}_(m-1)(x) \tag{8}$$

ii) Fit a base learner using the training set $\{x_i, -\frac{\hat{g}_m(x_i)}{\hat{h}_m(x_i)}\}_{i=1}^{N}$ by solving the optimization problem below:

$$\hat{\phi} = arg_{\phi \in \Phi} min \sum_{i=1}^{N} \frac{1}{2} \hat{h}_m(x_i)[-\frac{\hat{g}_m(x_i)}{\hat{h}_m(x_i)} - \phi(x_i)]^2 \tag{9}$$

$$\hat{f}_m(x) = \alpha \hat{\phi}_m(x) \tag{10}$$

iii) Update the model:

$$\hat{f}(x) = \hat{f}_{m-1}(x) + \hat{f}_m(x) \tag{11}$$

3. Output:

$$\hat{f}(x) = \hat{f}_{(M)}(x) = \sum_{m=0}^{M} \hat{f}_m(x) \tag{12}$$

*4.2.3 Inception-ResNet-v2.* Inception-ResNet-v2 is a variant of a convolutional neural network which is trained with images from the ImageNet database which over millions in amount. It is a deep layer network consisting of 164 layers and can perform image classification on thousands of categories of objects resulting in the learning of rich feature representations fora large variety of images in the network. The network has dimensions of image size of 299$X$299 [20].

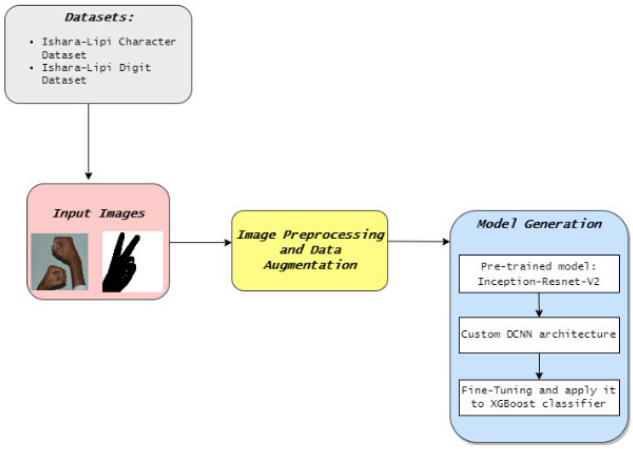


**Figure 4: The proposed procedure for BdSL recognition**

## 4.3 Optimization techniques

*4.3.1 Stochastic Gradient Descent.* Stochastic Gradient Descent is a very popular optimization technique when it comes to optimizing neural networks. It is mostly used for minimizing multi-variable objectives and differentiable function F(x). More specifically, it is the repetitive advanced technique of first order that can be used to find a differentiable function's local minima by taking steps which are iterative towards the location of F(x)'s location of the negative gradient. Its prime target is to reduce the loss function.

The following equation indicates a one-time update of a gradient descent (denoted by '$\nabla$' - represents the rate at which variable changes in relation to another variable. After a point's single update is received, $a_n$, it gives a point $a_{n+1}$ where $F(a_{n+1}) \leq F(a_n)$. $\gamma$ represents the rate of learning which defines the size step of each individual initiative which is in requirement of a meticulous choice of value.

$$a_{n+1} = a_n - \gamma \nabla F(a_n) \tag{13}$$

*4.3.2 Categorical crossentropy.* Categorical cross-entropy is a measure of the variance between the predicted and existing output distributions of a multi-class classifier. It is used to assess the performance of a model that is designed to classify items into one of several different classes. The loss amount is the size of the distinction between the probability distribution of both, and it is used to evaluate the model and guide its optimization.
The following equation is defined as the cross-entropy loss:

$$CE = -\sum_{i=1}^{C} t_i \log(s_i) \tag{14}$$

C is the total classes' value, $t_i$ is considered the ground truth and $s_i$ is the anticipated score of CNN.

The following equation is defined as overall cost:

$$J = \frac{1}{m} \sum_{i=1}^{m} CE_i \tag{15}$$

The goal of computing the cost of a model is to optimize its performance by fine-tuning the model parameters. The objective of training a model is to adjust its parameters so that it can better capture the training data's distribution. This is done by decreasing the categorical cross-entropy loss, which determines the discrepancy between the predicted and actual outputs produced by the model. By minimizing this loss, the model is able to better match the distribution of the training data and improve its performance. This is accomplished through an iterative process of training and parameter adjustment.

*4.3.3 Batch Normalization.* Batch normalization is a method used to improve the execution of deep neural networks. It involves normalizing the hidden layer's activation vectors utilizing the variance of the the set of data currently being processed and mean. This normalization step is performed just before the activation function is applied, which helps to stabilize the training process and make the network more efficient. By using batch normalization, deep neural networks can be trained more quickly and accurately.

$$\mu = \frac{1}{n} \sum Z^i \tag{16}$$

$$\sigma^2 = \frac{1}{n} \sum_i (Z^{(i)} - \mu)^2 \tag{17}$$

$$Z^{(i)}_{norm} = \frac{z^{(i)} - \mu}{\sqrt{\sigma^2 - \epsilon}} \tag{18}$$

$$= \gamma * Z^{(i)}_{norm} + \beta \tag{19}$$

*4.3.4 Max pooling.* Max pooling is a technique used in convolutional neural networks to reduce the resolution of the output from a layer. This has the effect of making the network consider larger areas of the input image at once, which reduces the amount of parameters that the network must learn and makes it more efficient to run. As a result, the computational load on the network is reduced.

*4.3.5 Dropout.* Dropout is a technique used during training in which some neurons are arbitrarily excluded from the network's calculations. This indicates that when the model is making predictions, the contribution of these neurons to the final output is ignored, and when the model is being trained, the weights of these neurons are not updated, permitting the model to centre its attention on the most important features and improve its performance over time. Neuron weights settle into their context within the network as a neural network learns. Neurons in a neural network are trained to recognize specific features in the input data, which allows them to specialize in detecting certain patterns. As a result, neighbouring neurons come to rely on this specialization, and the

overall model becomes more fragile. This can lead to overfitting, where the model is too specialized for the training data and performs poorly on new, unseen data. This reliance on context during training is known as complex co-adaptations [21].

*4.3.6 Grid search.* Grid Search Cross Validation is a technique for searching through the best parameter values in a grid of parameters. It is essentially a cross-validation technique. The model and its parameters must be entered. We use the tuned parameter values to build a new model using the training set after the Grid Search CV found the values for the hyperparameters. We can now evaluate our new model with the testing set [22].

## 5 EVALUATION

Accuracy is considered to be a very popular choice to determine the performance of image classification models. We evaluated our proposed model by using various other metrics such as precision, recall, and F-1 score alongside accuracy as accuracy alone is not enough to test a model's flexibility and can pose a challenge [23]. The equations for finding these metrics are given in Equations 20-23. To assess the model's performance, we need to compute several measures, such as true positive (TP), true negative (TN), false positive (FP), and false negative (FN). TP is a result in which the positive class is rightly predicted by the model whilst TN is when the model correctly produces the negative class. FP is when the positive class is wrongly forecasted by the model, whilst FN does the same for the negative class. These measures are used to assess the model's accuracy and inform its optimization.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{20}$$

$$Precision = \frac{TP}{TP + FP} \tag{21}$$

$$Recall = \frac{TP}{TP + FN} \tag{22}$$

$$F1 = 2 * \frac{(Recall * Precision)}{(Recall + Precision)} \tag{23}$$

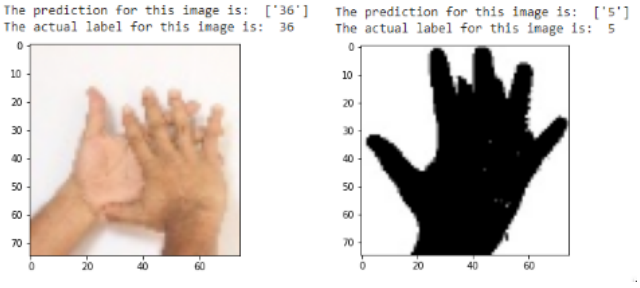


**Figure 5: Character and digit recognition after applying XGBoost classifier**

**Table 1: Accuracy attained in existing models**

| Existing BdSL Work | Dataset | Class | Accuracy (%) |
|---|---|---|---|
| Hasan and Ahsan (2019) | Ishara-Bochon: 1000 samples | 10 numerals | 94.74 |
| Ahmed et al. (2019) | 3200 samples | 10 numerals | 92 |
| Uddin et al. (2017) | 800 samples | 15 characters | 86.67 |
| Rahman et al. (2012) | 828 samples | 36 alphabets | 80.90 |
| Haque et al. (2019) | 130 samples | 26 alphabets | 77.88 |

## 6 EXPERIMENT

An ablation study on every component of our model was done and was evaluated empirically on the test dataset. We trained our model on Google Colab's K80 GPU at the backend with 200 epochs. The training was done in 2 hours on more than 55 million parameters and with a batch size of 50 images.

## 7 RESULTS

Figures 7 and 8 signify the loss and accuracy curve of the Deep-CNN model on the training data of Ishara-Lipi dataset. As we can see, the dataset is fitting properly to the model with both datasets. The batch size was taken to be 50 and the image size for both the datasets were 75x75x3. We had to train and test both the datasets separately. They were all trained for 200 epochs each. For the character dataset, we got an accuracy of 83.24% and for the digit dataset we got an accuracy of 93.33%. This DCNN model was then combined with the XGBoost classifier and this helped to increase the accuracy for character recognition to 86.67%, precision of 89%, recall and f1-score of 87% . For the digit recognition, we obtained score of accuracy of 97.33%, precision of 98%, recall and f-1 score of 97%. We even tested our model with other classifiers such as Random Forest and Decision Tree but XGBoost performed the best among all.

**Table 2: Performance on different classifiers**

| Our Proposed Model | Accuracy on Digit Dataset (%) | Accuracy on Character Dataset (%) |
|---|---|---|
| DCNN + Random Forest | 96.56 | 84.44 |
| DCNN + Decision Tree | 91.33 | 56.67 |
| **DCNN + XGBoost** | **97.33** | **86.67** |

Our proposed method also performs better than some already existing models such as Haque et al. (2019) which had an accuracy of 77.88% on the alphabets [4], Ahmed et al. (2019) which achieved 92% accuracy on the numerals [24], Uddin et al. (2017) with 86%

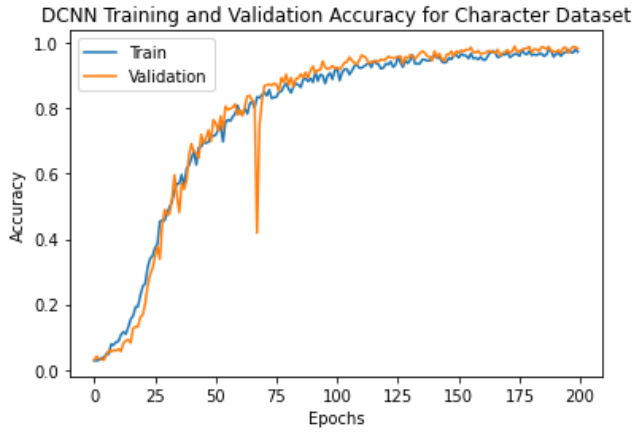


**Figure 6: Accuracy curve for DCNN training and validation stage on Character Dataset**

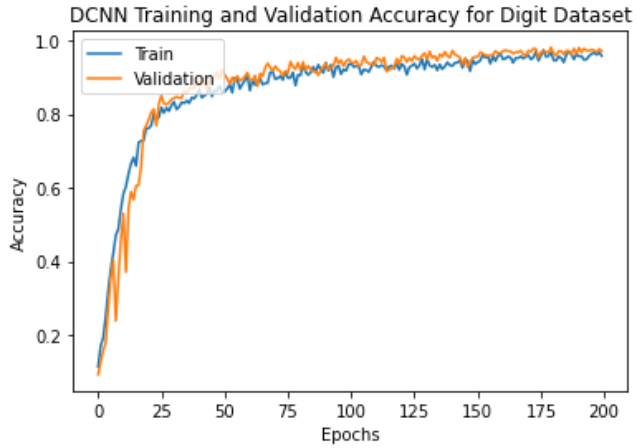


**Figure 7: Accuracy curve for DCNN training and validation stage on Digit Dataset**

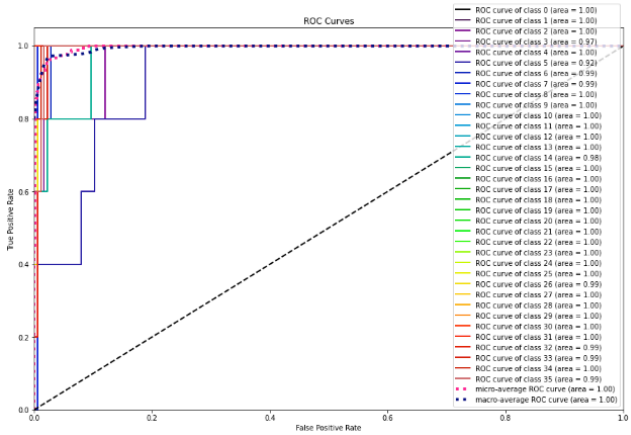


**Figure 8: ROC Curve of DCNN + XGBoost on Character Dataset**

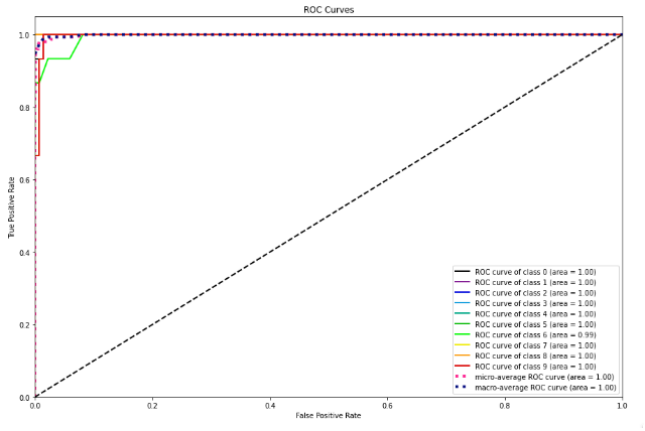


**Figure 9: ROC Curve of DCNN + XGBoost on Digit Dataset**

accuracy on the characters [6], 80.90% accuracy obtained by Rahman et. al (2012) on the Bengali alphabets and lastly [5], Hasan and Ahsan (2019) on the digit dataset achieved an accuracy of 94.74% [25]. The model created by us can correctly identify and recognize the sign language symbols with proper accuracy.

## 8 CONCLUSION

This paper describes a model for recognizing Bangla sign language that uses a deep convolutional neural network (DCNN) as its base. This model has been optimized by combining an Extreme Gradient Boosting (XGBoost) classifier with the proposed DCNN architecture. By combining these two techniques, we have been able to upgrade our model's performance and make it more effective at recognizing Bangla signs. For pre-processing, we have sharpened the image quality by using contrast limited adaptive histogram equalization and removing the background to reduce any unwanted noise which in turn aided in improving the model's performance. Due to having a small image dataset size, this model had to be adjusted accordingly by adding dropout of 20% in the DCNN architecture, batch sizes had to be small and had to hyper-tune XGBoost parameters by using grid search method. Thus, our model will give optimal results for any image classification project which has a relatively small dataset.

**Table 3: Accuracy attained in our proposed architecture**

| Our Proposed Model | Dataset | Class | Accuracy (%) |
|---|---|---|---|
| DCNN + XGBoost | Ishara-Lipi: 1005 samples for characters | 36 alphabets | **86.67** |
| DCNN + XGBoost | Ishara-Lipi: 1075 samples for digits | 10 numerals | **97.33** |

Our model has also outperformed quite a few recognition methods which are considered state-of-the-art. But due to having certain limitations like limited amount of data, and lack of a faster

GPU, we believe our model can yield even better results if these limitations are addressed. With more advanced pre-processing techniques, the model results can be improved further. In the future, we hope to make a real-time detection system of Bangla sign languages which can be used in any smartphone application.

## ACKNOWLEDGMENTS

To the martyrs of Bangla Language Movement of 1952, for sacrificing their lives on advocating the recognition of Bangla as the official language of Bangladesh (previously known as *East Pakistan*).

## REFERENCES

[1] WW Kong and Surendra Ranganath. Towards subject independent continuous sign language recognition: A segment and merge approach. *Pattern Recognition*, 47(3):1294–1308, 2014.

[2] Debashis Sarker. Living with mutism in the socio-economic context of bangladesh. *Wordgathering: A Journal of Disability Poetry and Literature*, 14(4), 2020.

[3] Sunanda Das, Md Samir Imtiaz, Nieb Hasan Neom, Nazmul Siddique, and Hui Wang. A hybrid approach for bangla sign language recognition using deep transfer learning model with random forest classifier. *Expert Systems with Applications*, 213:118914, 2023.

[4] Promila Haque, Badhon Das, and Nazmun Nahar Kaspy. Two-handed bangla sign language recognition using principal component analysis (pca) and knn algorithm. In *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pages 1–4. IEEE, 2019.

[5] Md Atiqur Rahman and Md Aktaruzzaman. Recognition static hand gestures of alphabet in asl. 2011.

[6] Jia Uddin, Fahmid Nasif Arko, Nujhat Tabassum, Taposhi Rabeya Trisha, and Fariha Ahmed. Bangla sign language interpretation using bag of features and support vector machine. In *2017 3rd International Conference on Electrical Information and Communication Technology (EICT)*, pages 1–4. IEEE, 2017.

[7] Chung-Lin Huang and Wen-Yi Huang. Sign language recognition using model-based tracking and a 3d hopfield neural network. *Machine vision and applications*, 10(5):292–307, 1998.

[8] Rahat Yasir and Riasat Azim Khan. Two-handed hand gesture recognition for bangla sign language using lda and ann. In *The 8th international conference on software, knowledge, information management and applications (SKIMA 2014)*, pages 1–5. IEEE, 2014.

[9] Farhad Yasir, PW Chandana Prasad, Abeer Alsadoon, and Amr Elchouemi. Sift based approach on bangla sign language recognition. In *2015 IEEE 8th international workshop on computational intelligence and applications (IWCIA)*, pages 35–39. IEEE, 2015.

[10] Muttaki Hasan, Tanvir Hossain Sajib, and Mrinmoy 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)*, pages 1–5. IEEE, 2016.

[11] Md Tazimul Hoque, Md Rifat-Ut-Tauwab, Md Fasihul Kabir, Farhana Sarker, Mohammad Nurul Huda, and Khandaker Abdullah-Al-Mamun. Automated bangla sign language translation system: Prospects, limitations and applications. In *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, pages 856–862. IEEE, 2016.

[12] Yunus Can Bilge, Nazli Ikizler-Cinbis, and Ramazan Gokberk Cinbis. Zero-shot sign language recognition: Can textual data uncover sign languages? *arXiv preprint arXiv:1907.10292*, 2019.

[13] Hovannes Kulhandjian, Prakshi Sharma, Michel Kulhandjian, and Claude D'Amours. Sign language gesture recognition using doppler radar and deep learning. In *2019 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6. IEEE, 2019.

[14] Md Mehedi Hasan, Azmain Yakin Srizon, and Md Al Mehedi Hasan. Classification of bengali sign language characters by applying a novel deep convolutional neural network. In *2020 IEEE Region 10 Symposium (TENSYMP)*, pages 1303–1306. IEEE, 2020.

[15] Samiya Kabir Youme, Towsif Alam Chowdhury, Hossain Ahamed, Md Sayeed Abid, Labib Chowdhury, and Nabeel Mohammed. Generalization of bangla sign language recognition using angular loss functions. *IEEE Access*, 9:165351–165365, 2021.

[16] Md Sanzidul Islam, Sadia Sultana Sharmin Mousumi, Nazmul A Jessan, AKM Shahariar Azad Rabby, and Sayed Akhter 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)*, pages 1–4. IEEE, 2018.

[17] Md Shafiqul Islalm, Md Moklesur Rahman, Md Hafizur Rahman, Md Arifuzzaman, Roberto Sassi, and Md Aktaruzzaman. Recognition bangla sign language using convolutional neural network. In *2019 international conference on innovation and intelligence for informatics, computing, and technologies (3ICT)*, pages 1–6. IEEE, 2019.

[18] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[19] Rafael C Gonzalez. *Digital image processing*. Pearson education india, 2009.

[20] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[21] Jason Brownlee. Dropout regularization in deep learning models with keras, Aug 2022.

[22] Wei-Meng Lee. Tuning the hyperparameters of your machine learning model using grid search cv, 2021.

[23] Haoyang Xu. The problem with'accurate'history: Complexity within sallust's bellum catilinae. *Int'l J. Soc. Sci. Stud.*, 8:81, 2020.

[24] Shahjalal Ahmed, Md Islam, Jahid Hassan, Minhaz Uddin Ahmed, Bilkis Jamal Ferdosi, Sanjay Saha, Md Shopon, et al. Hand sign to bangla speech: a deep learning in vision based system for recognizing hand sign digits and generating bangla speech. *arXiv preprint arXiv:1901.05613*, 2019.

[25] Md Mahmudul Hasan and Sk Mohammad Masudul Ahsan. Bangla sign digits recognition using hog feature based multi-class support vector machine. In *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*, pages 1–5. IEEE, 2019.