BdSL47: A Complete Open-access Depth-based Bangla Sign Alphabet and Digit Dataset

S M Rayeed[1*], Sidratul Tamzida Tuba[1†], Hasan Mahmud[1†], Md. Mumtahin Habib Ullah Mazumder[2†], Md. Saddam Hossain Mukta[2†], Md. Kamrul Hasan[1†]

[1] Systems and Software Lab (SSL), Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Board Bazar, Gazipur, 1704, Dhaka, Bangladesh.

[2] Department of Computer Science and Engineering (CSE), United International University (UIU), United City, Madani Avenue, Dhaka, 1212, Bangladesh.

[*]Corresponding author's email : rayeed@iut-dhaka.edu

Contributing authors' email : sidratultamzida@iut-dhaka.edu ; hasan@iut-dhaka.edu ; mumtahin@cse.uiu.ac.bd ; saddam@cse.uiu.ac.bd ; hasank@iut-dhaka.edu

[†] These authors contributed equally to this work.

Abstract. Sign Language Recognition (SLR) targets on interpreting the sign language into text or speech, in order to facilitate the communication between deaf-mute people and ordinary people. Constructing a sign language dataset has broad social impact, but is challenging due to its complexity and large variations in hand actions. In Bangla Sign Language (BdSL), not many of such datasets are available online, and the existing ones are based on RGB images, whereas, having a depth dataset to incorporate depth information during the classification phase can significantly increase the recognition accuracy. In this paper, we have proposed a complete depth-based open-access dataset for 47 one-handed static signs (10 digits, from ০ to ৯; and 37 letters, from অ to ঁ) of Bangla Sign Language, using the MediaPipe framework for extracting depth information. For classification, we primarily used baseline classifiers, later we designed an Artificial Neural Network (ANN) model that resulted in an accuracy of 97.78%. Moreover, we have compared the Bangla sign alphabet dataset with an existing benchmark dataset ‘Ishara Lipi’ using deep learning based approach and achieved 98.92% recognition accuracy, which is significantly higher than the Ishara Lipi dataset.

Keywords: Sign Language Recognition, Bangla Sign Language, Bangla Sign Alphabet Dataset, Depth Information, Hand Key-points, Hand Landmark Model, MediaPipe.

1 Introduction

Sign language is a non-verbal form of communication used by deaf and hearing impaired people, in order to communicate with others through bodily movements e.g. hand gestures, facial expression and body language. It is the primary mode of communication for them, yet in many cases, incomprehensible to many; and that's where machine learning applications turn out to be utilitarian. Pre-trained models can help ease interpreting what a person is trying to express, therefore easing the communication. This recognition process is called Sign Language Recognition (SLR), which is a section of gesture recognition, one of the main focuses of computer vision research in recent times. The main goal of gesture recognition is to build a software or a system that is capable of detecting and identifying particular human gestures, based on the mathematical analysis of the gestures, in order to perform actions. Hand signs in sign languages are a collection of specified hand gestures, so several computer vision based solutions applied in gesture recognition are becoming popular for sign language recognition.

In Bangladesh, almost 2.6 million deaf and hearing impaired people depend on sign language to communicate in their day to day life, which substantiates the necessity of a digital communication system (Islam et al., 2018). For extensive research on sign language recognition, the need for a complete open-access dataset is paramount. In terms of building open-access datasets for BdSL, it has got off to a start with Islam et al. in 2018, who published two separate datasets: 'Ishara Bochon' for sign digits (Islam et al., 2018). and 'Ishara Lipi' for sign characters (Islam et al., 2018). In both cases, RGB images were used for dataset generation, and Convolutional Neural Network (CNN) was used for classification. Another state-of-the-art dataset is BdSL36 (Hoque et al., 2020), a Bangla sign alphabet dataset which incorporates background augmentation to make the dataset versatile and contains over four million images belonging to 36 categories. These datasets contain 36 two-handed static signs of Bangla sign alphabet, but according to the Bangla Sign Language Dictionary, there are 37 static one-handed signs as well[1]. Most importantly, none of the existing datasets in Bangla Sign Language contains depth information, whereas several research works on other sign languages have shown that, instead of just relying on RGB images, extracting depth information from a given hand shape and using it for classification significantly improves the classification accuracy (Doliotis et al., 2011). Depth information can be extracted from depth-sensors, e.g. Intel RealSense, Microsoft Kinect. These sensor-provided depth values can be used as spatial hand-shape features in sign language recognition. However, such devices are very expensive. An efficient low-cost alternative is the framework named MediaPipe, a relatively recent addition to the field of applied machine learning, specialized in face detection, hand key-points detection, multi-hand tracking and hair segmentation (Zhang et al., 2020). The MediaPipe Hand module is designed to detect 21 predefined hand key-points, delineate a geometric representation of the hand skeleton, provide normalized 2D coordinate values and estimate a depth value of these points using depth map, from RGB images with no

[1] Link : https://dokumen.tips/embed/v1/bangla-sign-language-dictionary.html

extra devices. The main advantage of using this framework is that it is capable of extracting depth information without the need of an extra depth-sensor. However, the extracted depth values are estimated, as opposed to accurate sensor-provided depth values from aforementioned sensing devices. But still these estimated values can effectively be used as spatial features and provide higher accuracy; as shown in our previous work on Bangla sign digits (Rayeed et al., 2022), using the MediaPipe framework, we achieved an accuracy of 98.65%.

In this paper, we have presented BdSL47: a complete depth-based open-access dataset of 47 one-handed static signs of sign alphabet and sign digits of Bangla Sign Language using depth information via MediaPipe. This is the first open-access complete dataset in Bangla Sign Language to contain both sign digits and sign alphabet. First of all, we have constructed the bangla sign alphabet dataset using depth information, and then merged it with the previous sign digit dataset. In the recognition phase, we have initially evaluated the performance using three classical machine learning models, K-Nearest Neighbor (KNN), Support Vector Machine (SVM) and Random Forest Classifier (RFC). Additionally, we have designed an ANN architecture, and conducted separate classification for the sign alphabet dataset and the combined dataset, tuning the ANN model accordingly, both of which gave higher accuracy than that of state-of-the-art research works (Islam et al., 2018; Islam et al., 2018; Hoque et al., 2020). We have considered Ishara Lipi as our reference dataset. For an equivalent comparison of the datasets, we have used the RGB samples from our alphabet dataset and applied the same CNN architecture as Ishara Lipi on that dataset, which yielded a better accuracy. Figure 1 shows images of 47 signs (10 digits, from ০ to ৯; and 37 letters, from অ to ঁ) of the proposed BdSL47 dataset :

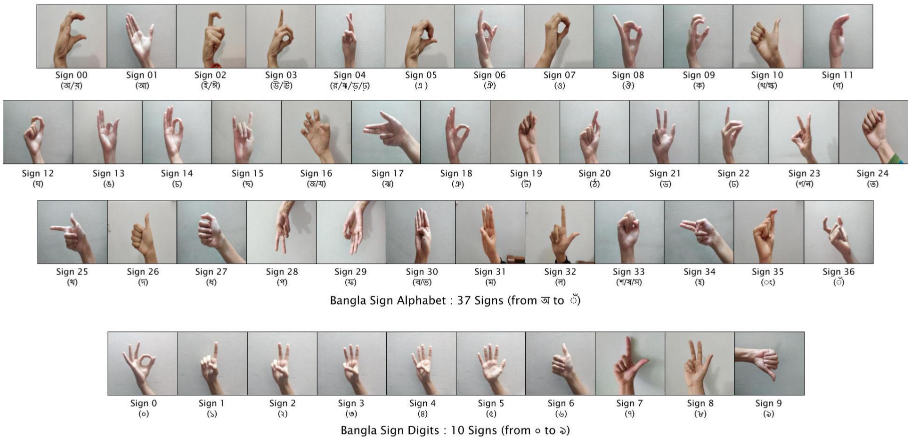


Fig. 1. Image samples of 47 signs from BdSL47 dataset

Our contributions in this paper are summarized as follows :

1. To the best of our knowledge, the proposed ‘BdSL47’ dataset is the first complete depth-based dataset in Bangla Sign Language, containing both sign digits and sign alphabet. The dataset consists of one-handed static hand gestures of total 47 signs, 37 of sign alphabet (from sign অ to sign ঁ), and 10 of sign digits (from sign ০ to sign ৯); hence, 47000 input images, 47000 corresponding output images and 470 csv files (containing the coordinate values of the key-points).
2. For recognition purposes, along with baseline machine learning classifiers (KNN, SVM, RFC), we have designed an ANN architecture, and used the model on BdSL47 dataset that resulted in an accuracy of 97.78%. We have also conducted classification on the sign alphabet dataset using the same ANN model (with necessary parameter changes), which yielded an accuracy of 99.41%.

2 Literature Review

Sign language involves non-vocal communication with a combination of hand gestures, lip patterns and facial expressions. Historically the origination of sign language dates back to the 16th century, when an Italian physician, Geronimo Cardano, realized the necessity of a common language to help the deaf community communicate with the world. The first book on sign language was published in 1620, and by the next century, the first sign language school was established. In the late twentieth century, research on sign language got propelled. At first the sign language character recognition was mostly conducted based on RGB images, but since last decade, depth information has been incorporated in many of the research works, because of its significant impact on increasing the accuracy (Doliotis et al., 2011; Liang and Yuan, 2014).

2.1 Ishara Lipi : First multi-purpose open access Bangla Sign Alphabet Dataset (Islam et al., 2018)

In 2018, Islam et al. published the first multi purpose open access Bangla sign alphabet dataset, named Ishara Lipi (Islam et al., 2018). The dataset contains 1800 RGB image samples of 36 two-handed static signs of the Bangla sign alphabet, 6 vowels and 30 consonants, collected from different deaf and general volunteers. After collecting raw RGB images, data preprocessing, filtering and segmentation were performed for recognition purposes. The dataset generation of Ishara Lipi is divided into following states -

1. Sample Capturing: The dataset contains 1800 images of 36 signs from Bangla sign alphabet. First, RGB image samples of uncovered hands were captured in a plain white background.
2. Data Labeling: The authors labeled 36 characters with numeric convention from 1 to 36. After data labeling, there were 50 RGB samples in each category.
3. Image Cropping: The samples were cropped in order to show the region of interest (user-hand) observing the rate of height and width for further processing and better feature extraction.
4. Image Resizing and Converting to Grayscale: For easier and more accurate classification, images were resized into 128×128 pixels, then converted from RGB to grayscale.

Overall process of data collection and classification is given in Fig. 2:

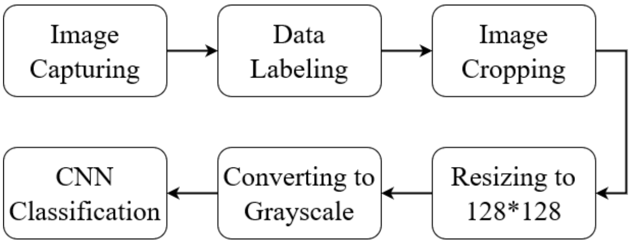


Fig. 2. Overview of Ishara Lipi: Dataset Generation and Classification

After data preprocessing, the images were fitted into a 9-layer CNN classification model, having 2 sets of Convolutional and Max Pooling Layer. Adam optimizer was used for optimization, with a learning rate of 0.001%. The model achieved 94.74% validation accuracy and 92.65% training accuracy (Islam et al., 2018).

2.2 Other works on Bangla Sign Language

Apart from Ishara-Lipi, there have been several research studies conducted on the sign alphabet of Bangla Sign Language. In 2015, a computer vision-based approach was proposed using contour analysis and Haar-like feature-based cascaded classifiers, which yielded 90.11% recognition accuracy on a dataset of 1800 contour templates for 18 Bangladeshi sign words separately (Rahman et al., 2015). In 2018, the same authors presented a real-time hand-signs segmentation and classification system using fuzzy rule based RGB model and grid-pattern analysis (Rahman et al., 2018). The system was tested on a dataset of 46 hand-signs of Bengali sign language (BdSL) and 10 hand-signs of Chinese sign language (ChSL), consisting of a total number of 33,600 testing images in six different backgrounds with illumination variation environments from 10 new signers. The proposed system achieves mean accuracy of 95.67% for BdSL and 96.57% for ChSL with the computational cost of 8.01 milliseconds per frame in six challenging environments. The authors further extended their research on Bangla sign language by developing Bangla Language Modeling Algorithm (BLMA): a method of recognizing hand-sign-spelled Bangla language (Rahman et al., 2020). They proposed a two-step classifier for hand-sign classification using normalized outer boundary vector (NOBV) and window-grid vector (WGV) by calculating maximum inter correlation coefficient between test feature vector and pre-trained feature vectors. The system is trained using 5200 images and tested using another (5200×6) images of 52 hand-signs from 10 signers in 6 different challenging environments, achieving a mean accuracy of 95.83% for classification with the computational cost of 39.972 milliseconds per frame. Ahmed and Akhand proposed a method of calculating relative tip positions of five fingers in two dimension space and used the position vectors to train an Artificial Neural Network (ANN) which yielded an accuracy of 99% on a dataset of 518 images of 37 signs (Ahmed and Akhand, 2016). Hoque et al. implemented a real-time sign detection system using Faster R-CNN, which performed with 98.2% accuracy on a dataset of 10 different bangla sign letters (Hoque et al., 2018). Later, the authors extended their work by constructing a complete dataset of 36 signs; consisting of more than 470000 images, assembled from only 1200 original samples of 36 two-handed signs using data augmentation (Hoque et al., 2020). For classification, they used deep learning techniques, such as ResNet34, ResNet50, VGGNet19, Densenet169, Densenet20, and achieved the maximum accuracy of 99.10% using VGGNet19.

The major limitation of these works is none of these datasets were constructed using depth information, which can improve classification accuracy. Also most of them are not available for further research. These datasets are mostly based on two-handed signs, unlike the one handed signs of our proposed dataset. Moreover, previous datasets are image datasets, whereas, our dataset is primarily a tabular dataset, although we have an image dataset of 47000 samples as well. Despite these incompatibilities, for a better overview of our dataset, we have considered the Ishara-Lipi (Islam et al., 2018) dataset as our reference dataset and compared this dataset with our image dataset.

2.3 Depth Information in Sign Language Recognition (SLR)

Research on sign language recognition was initially based on RGB images, however, in recent research works, depth information has been incorporated for achieving better accuracy since it offers some additional advantages, such as appending multimodality and automated feature learning, and enabling depth quantization for fine gestures; which significantly increases the recognition accuracy.

In 2011, while comparing the accuracy in hand gesture recognition between depth and RGB dataset, it was found that the depth dataset using Microsoft Kinect outperforms the RGB dataset, in cases of both static and complex backgrounds (Doliotis et al., 2011). Later that year, a hand-shape recognition system using Microsoft Kinect and OpenNI+NITE framework was proposed, for detecting and tracking the hand from depth images (Pugeault and Bowden, 2011). In 2013, Rodriguez and Chavez proposed a hand gesture recognition model based on hand segmentation using depth map and feature extraction from segmented samples using Scale-Invariant Feature Transformation

(SIFT). Applying the model on the American Sign Language dataset and classifying via SVM, an accuracy of 90.2% was recorded. (Rodriguez and Chavez, 2013). A year later, a distance-adaptive feature selection method was proposed for extracting discriminative depth-context features for hand recognition, increasing accuracy by 17.2% (Liang and Yuan, 2014). In later research works, depth sensing devices became popular for generating depth images and extracting depth information (Almasre and Al-Nuaim, 2016; Mudduluru, 2017; Liao et al., 2018). In Indian Sign Language digit recognition, Intel RealSense was used to extract depth information of 22 hand-joint points from sign digits, which were input features of SVM classification with an accuracy of 93.5% (Mudduluru, 2017). A supervised machine learning model was proposed for hand gesture recognition on Arabic Sign Language using Microsoft's Kinect with Leap Motion Controller, which significantly increased the accuracy (Almasre and Al-Nuaim, 2016). In 2018, Liao et al. proposed a hand gesture recognition model based on generalized Hough transform, using Intel RealSense, by mapping depth images to color images for hand segmentation in complex backgrounds. For classification, a Double-Channel Convolutional Neural Network (DC-CNN) model was proposed, with dedicated channels for RGB and depth images, providing an accuracy of 99.4% on the English Sign Alphabet dataset (Liao et al., 2018). Al Marouf et al. used Microsoft Kinect in fingertip detection and finger identification for Hand Gesture Recognition, which resulted in an accuracy of 94% (Al Marouf et al., 2018). For symbolic gesture recognition using depth information, Mahmud et al. used Scale-Invariant Feature Transform algorithm, which takes the generated depth silhouettes as input and produces robust feature descriptors as output (Mahmud et al., 2018). The features were classified using multiclass SVM and yielded 96.84% accuracy.

2.4 Use of MediaPipe Framework

For most of the research works mentioned in the previous section, depth sensing devices have been used for extracting depth information, which are quite expensive. A low-cost alternative of extracting depth information from RGB images is the MediaPipe framework. The framework offers more affordability, accessibility, comfortability and ease of usage, as it requires no additional sensors. Cutting edge machine learning models using MediaPipe include face detection, hand key-points detection, multi-hand tracking, hair segmentation, 3D object detection and tracking. The module used for hand gesture recognition is MediaPipe Hands (See Figure 3), consisting of two underlying models – Palm Detection Model and Hand Landmark Model, and providing a high-fidelity solution for hand and finger tracking.

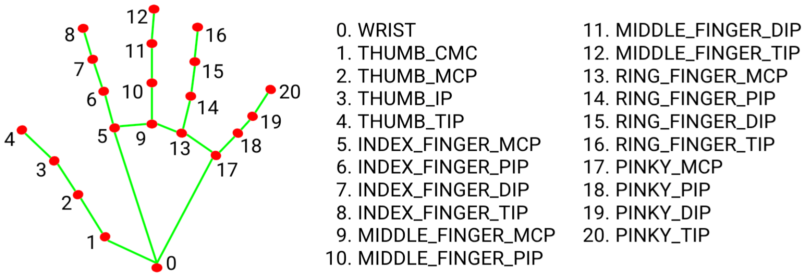


Fig. 3. Predefined 21 Hand Key-points of MediaPipe Hand Module

The former model operates on the whole image to detect the user hand(s), while the latter operates on the cropped image region defined by the former and returns high-fidelity 3D hand key-points (Zhang et al., 2020). Though the framework is relatively recent, several research works have been conducted using MediaPipe (Halder and Tayade, 2021; López, 2020; Ghosh, 2021; Herath and Ishanka, 2022; Harris and Agoes, 2021). Halder and Tayade used MediaPipe on American, Indian, Italian and Turkish sign language datasets to analyze the performance of the framework, which yielded an average accuracy of

99%. López used MediaPipe for recognizing four different gestures of American Sign Language, where Recurrent Neural Networks (RNN) yielded a classification accuracy of 92%. In 2021, Ghosh proposed a Keras RNN-LSTM model using MediaPipe for real-time detection of 5 words from American Sign Language and achieved an accuracy of 95%. Harris and Agoes proposed a mock-up user guide application, using MediaPipe for hand gesture recognition, for better user interaction and utilization convenience. The classification accuracy was 95% on a dataset of 900 samples from 10 different hand gestures. Herath and Ishanka proposed an approach to Sri Lankan sign language recognition using deep learning with MediaPipe, where they achieved over 95% accuracy using LSTM, CNN and CNN-LSTM.

3 Proposed Approach

This section contains the steps of generating the BdSL alphabet dataset using the MediaPipe framework. To construct the BdSL alphabet dataset from scratch, we had to go through several phases, which broadly can be categorized into (1) data collection and preprocessing and (2) classification. However, a detailed approach involves these following steps.

3.1 Image Sample Collection

The first step of constructing the BdSL alphabet dataset was to collect RGB image samples. For every sign (from sign অ to sign ঁ) from a user, more than 150 frames were captured, from which only 100 that could most accurately detect the hand key-points were finalized. Taking into account that some frames may yield faulty or no hand detection due to darker lighting ambience, blurry hand-signs, lack of focus and hand-movement while capturing, some extra samples were taken. The samples were collected via general-purpose webcam. Our primary focus was to capture the user-hand, hence, while taking samples, only the user-hand was in the field of view of the camera, and therefore no explicit hand segmentation was used. Figure 4 shows the collection of samples from a user for a particular sign (sign 14 - চ) in a folder.

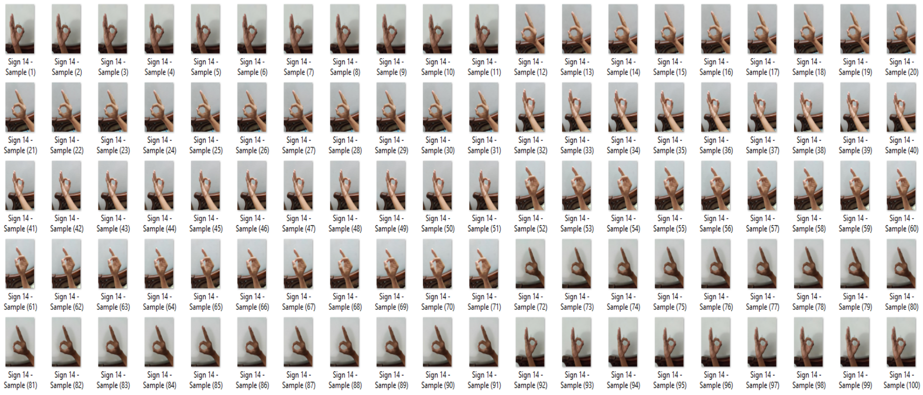


Fig. 4. Collection of captured of samples (sign 14 - চ) in a specified folder

Variation in sample collection. To evaluate the accuracy of the MediaPipe framework to perform hand key-points detection, the image samples were collected with user variation and environment variation being taken into account. Among the 10 users participating in the dataset-generation process, variation occurred in terms of expertise, age, gender, hand-shape and skin color. Users were different from each other in at least one of the factors. We also have considered environmental variations in different aspects like scaling, translation, hand-rotation, hand-orientation, lighting ambience, background. Some examples of such variations are shown in Figure 5.

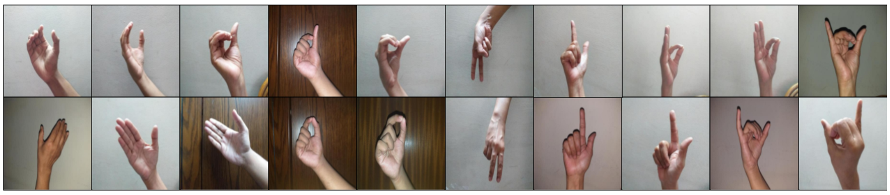


Fig. 5. Variation in User and Environment Setup during sample collection

3.2 Hand Key-points Detection

After capturing the RGB images, the image frames were processed via the MediaPipe framework for hand-tracking and hand key-points detection. MediaPipe, as aforementioned, runs two underlying models – the palm detection model, BlazePalm, for hand detection, and Hand

Landmark Model for 21 hand key-points detection. The image samples were processed through MediaPipie individually. Fig. 6 depicts how we have used the MediaPipe framework for detecting hand keypoints and generating a corresponding output image for each sample.

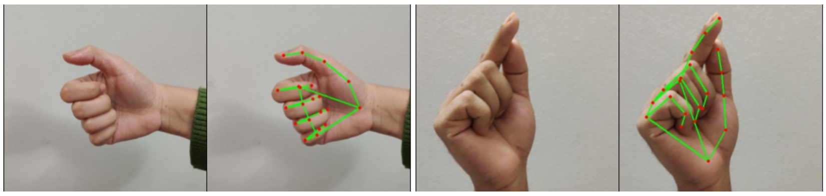


Fig. 6. Input Images with Corresponding output images

3.3 Removal of Samples with Faulty Key-points Detection

One of the limitations we have faced while working with MediaPipe is, in case of few samples, processing via MediaPipe sometimes resulted in faulty detections of hand keypoints, mostly due to darker lighting ambience, user-hand being blurred and complex background. Because of this, after generating the output images, we had to manually check the samples for such faulty cases to discard them. After scrutinization, 100 samples that accurately detected hand key-points were selected for each sign from every signer. Some examples of such variations are shown in Figure 7.

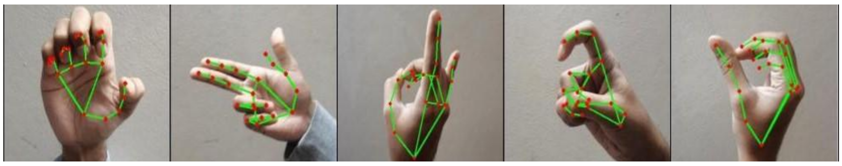


Fig. 7. Discarded output samples with Faulty hand key-points detection

3.4 Data Files Generation

After manual checking and selection, the image samples were resized into a standard size, 640×480 in case of landscape images and 480×640 in case of portrait images. After a final processing of these images via MediaPipe, we stored the normalized x, y and z values for each of the 21 hand keypoints detected from a sample. For normalizing the values of x and y coordinates, the MediaPipe framework uses min-max normalization where all the values are scaled in the range from 0 to 1. For storing depth values, standardization has been used with a mean of 0 and standard deviation of 1, where the smaller the value the closer the landmark is to the camera. For 100 samples of a sign from a user, we created one csv file, consisting of the normalized 3D coordinate values of 21 hand key-points, having 65 columns (including name and label). A sample csv file is given below in Table 1:

Table 1. Sample csv file (**First Column:** Sample name; **Columns x_{00}, y_{00}, z_{00}, …, x_{20}, y_{20}, z_{20} :** 3D values of 21 hand keypoints; **Last Column**: Label)

Name	x_{00}	y_{00}	z_{00}	…	…	x_{20}	y_{20}	z_{20}	Label
1.jpg	0.5268	0.7077	-0.0015	…	…	0.6331	0.3030	-0.0290	5
1.jpg	0.5248	0.6950	-0.0017	…	…	0.6325	0.2845	-0.0215	5
1.jpg	0.5299	0.7029	-0.0017	…	…	0.6290	0.2685	-0.0353	5
…	…	…	…	…	…	…	…	…	5
…	…	…	…	…	…	…	…	…	5
98.jpg	0.4833	0.7143	-0.0037	…	…	0.3734	0.3343	-0.0637	5
99.jpg	0.4827	0.7285	-0.0064	…	…	0.3590	0.3404	-0.0297	5
100.jpg	0.4894	0.7471	-0.0060	…	…	0.3514	0.3380	-0.0384	5

From the table above, we can see that the x, y, z coordinate values are stored in separate 63 columns, these columns are the skeleton of our tabular dataset. Since a csv file contains information about 100 images of a sign collected from one user, there are 100 rows (without heading).

3.5 Constructing BdSL47 : Merging the Sign Alphabet Dataset with previous Sign Digit Dataset

As mentioned earlier, in our previous work on Bangla sign digits, we constructed a complete dataset of 10 signs of Bangla sign digits. After completion of the sign alphabet dataset, we have merged the two datasets, which now consists of 47 signs (10 digits and 37 letters) and has been labeled as ‘BdSL47’. This is the first dataset of Bangla Sign Language (BdSL) to contain both sign digits and sign alphabet.

However, while merging two datasets, we had to do some basic adjustments in labeling. In the alphabet dataset, we have labeled the samples from 0 to 36 for 37 signs of Bangla sign alphabet, and in the BdSL digit dataset, the images were also labeled from 0 to 9 for the 10 digit signs; therefore, we had to relabel the samples from sign digits for classification purpose in the merged dataset to avoid mislabeling which would significantly decrease the accuracy. The 10 sign digits (from sign ০ to ৯) have been relabeled from 37 to 46, following Table 2.

Table 2. Relabeling of sign digits from BdSL digit dataset

Sign Digit	Label in Digit Dataset	Label in BdSL47	Sign Digit	Label in Digit Dataset	Label in BdSL47
0 (০)	0	37	5 (৫)	5	42
1 (১)	1	38	6 (৬)	6	43
2 (২)	2	39	7 (৭)	7	44
3 (৩)	3	40	8 (৮)	8	45
4 (৪)	4	41	9 (৯)	9	46

The overall process of the proposed approach is given in the following Flow Diagram in Figure 8.

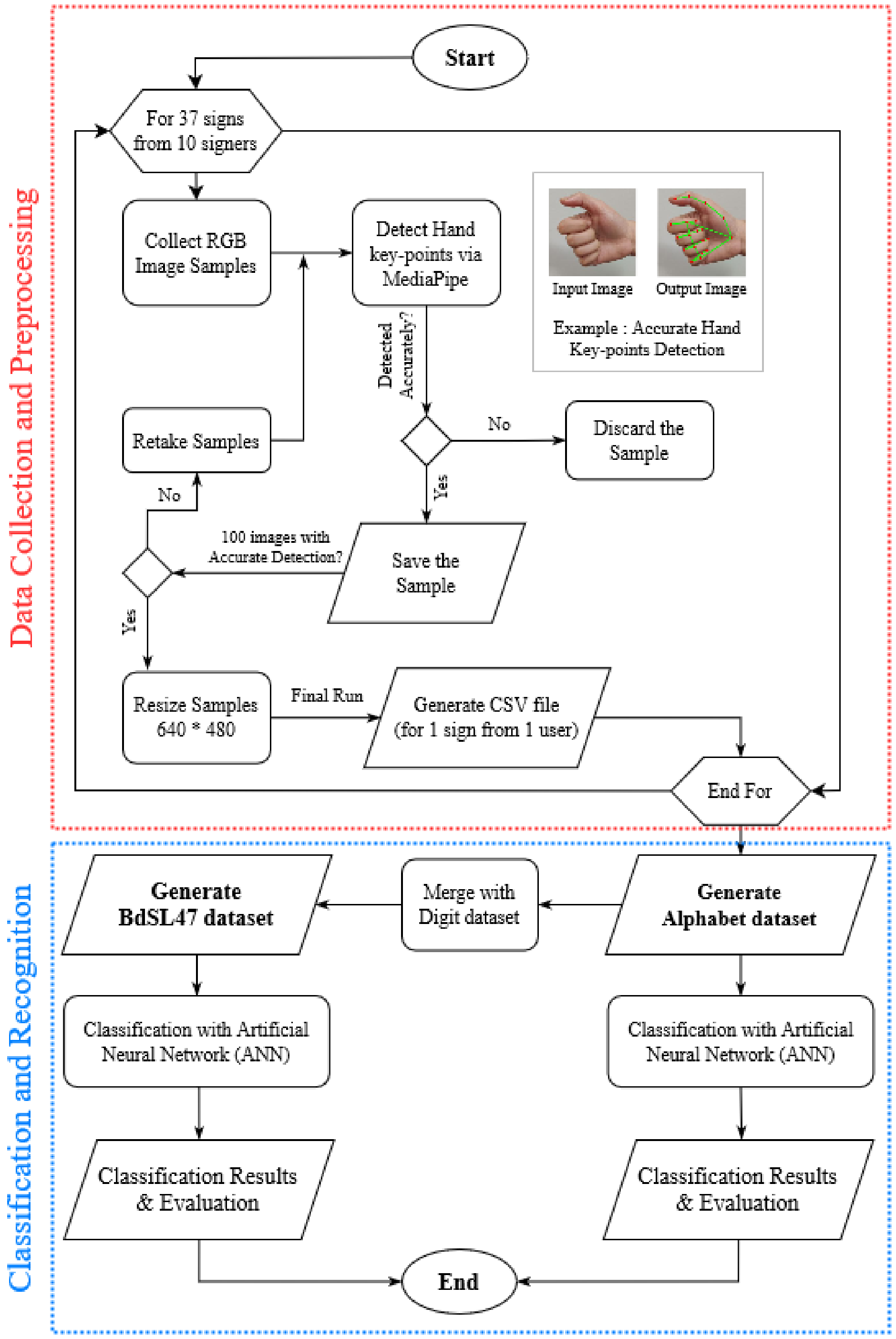

Data Collection and Preprocessing
Start
For 37 signs from 10 signers
Collect RGB Image Samples
Detect Hand key-points via MediaPipe
Input Image
Output Image
Example : Accurate Hand Key-points Detection
Detected Accurately?
No
Discard the Sample
Retake Samples
Yes
No
100 images with Accurate Detection?
Save the Sample
Yes
Resize Samples 640 * 480
Final Run
Generate CSV file (for 1 sign from 1 user)
End For
Classification and Recognition
Generate BdSL47 dataset
Merge with Digit dataset
Generate Alphabet dataset
Classification with Artificial Neural Network (ANN)
Classification with Artificial Neural Network (ANN)
Classification Results & Evaluation
Classification Results & Evaluation
End

Fig. 8. Flow Diagram of Dataset Generation and Classification process

The proposed BdSL47 dataset, generated and analyzed during the current research study, is publicly available in the Harvard Dataverse repository, https://doi.org/10.7910/DVN/EPIC3H (Rayeed, 2022).

Statistical information of BdSL47 dataset is given below in Table 3.

Table 3. Dataset Statistics of the proposed BdSL47 Dataset

No. of input images	47000	Image size	640 × 480
No. of users	10	Sign type	Static One handed
No. of total signs	47	No. of csv files	470
No. of alphabet signs	37 (অ to ◌ঁ)	No. of samples per csv	100
No. of digit signs	10 (০ to ৯)	No. of input features	63
Image type	RGB	No. of output labels	47

4 Classification and Analysis

In this research study, to give a baseline of our proposed dataset, we have performed several classifications on our proposed dataset and analyzed the results.

Firstly, we have performed classification on the sign alphabet dataset using KNN, SVM and RFC; however the accuracy was not satisfactory, because the classical machine learning models are not optimized to perform automated feature learning unlike neural networks. So, we have designed an Artificial Neural Network model to evaluate its performance on the alphabet dataset. Then, we have used the similar approach on the merged BdSL47 dataset. Keeping the core architecture as it is, we did few parameter changes to analyze its performance on the proposed BdSL47 dataset.

The first two classifications have been conducted on the tabular dataset. However, the third classification is for an equivalent comparison of the image samples, to assess the image dataset. For comparison with our reference dataset Ishara Lipi, we took the image samples from the sign alphabet dataset, and did similar preprocessing as Ishara Lipi to make the samples homologous to the images of the reference dataset. Then, we performed CNN-based classification on our dataset, with the same CNN architecture used for classifying the Ishara Lipi dataset and compared the classification results of both the datasets.

4.1 Classification on the sign alphabet dataset

The sign alphabet dataset consists of 37000 RGB image samples for 37 static one-handed signs, collected from 10 signers. After processing via MediaPipe, 370 corresponding csv files have been generated.

These 370 csv files form the tabular dataset, on which the ANN model has been run. The csv files contain x, y and depth coordinate values of 21 predefined hand key-points extracted from the samples. As these values have been used as the input features, there are 21×3, or 63 input features. It is a multilabel classification problem, with 37 output labels. Prior to classification, 10 fold cross validation was used and the dataset was splitted into a standard train test split ratio of 80 : 20. Initially, we have performed classification on the alphabet dataset, using KNN, SVM and RFC. Table 4 shows the accuracy results.

Table 4. Accuracy Results for KNN, SVM, RFC on the alphabet dataset

Classifier	Accuracy (%)
KNN	84.22
SVM	86.70
RFC	92.26

As the classification results were not satisfactory, for better accuracy, we have architected an ANN classification model, with 21×3=63 nodes in the input layer, 37 nodes in the output layer (for 37 letters of bangla sign alphabet), and 4 hidden layers with a dropout rate of 33% and 25% in the last two hidden layers respectively.

The architectural diagram of the ANN model is shown in Figure 9.

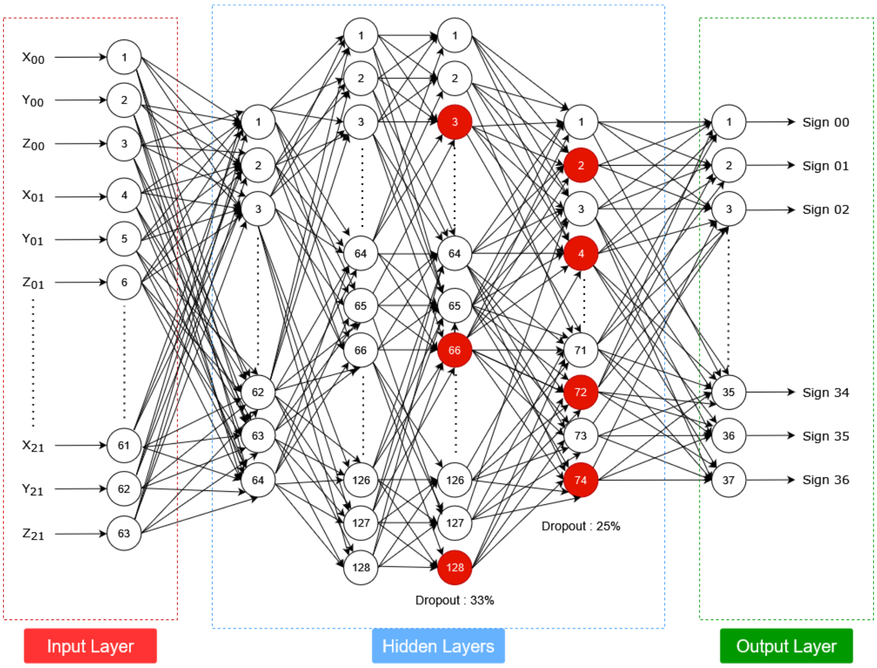


Fig. 9. Architectural Diagram of the ANN model for sign alphabet dataset

Optimizer, Learning Rate and Loss Function. The accurate choice of optimization algorithm significantly impacts on the classification results in machine learning and computer vision applications. Among the contemporary deep learning optimizers e.g. Adam, Stochastic Gradient Descent (SGD), AdaGrad, RMSProp, the first one suits the best in our case. The Adam optimization algorithm is an upgraded extension of SGD which, unlike SGD, does not maintain a constant learning rate through training, but learns to adapt and updates learning rate for each network weight individually. It can also be viewed as an extension of the RMSProp, because it not only relies upon the first moment for adapting learning rates like RMSProp, but it also uses the second moment of gradients (Kingma and Ba, 2014). For its ubiquity, efficiency and accuracy, it has been adapted as a benchmark for deep learning papers and recommended as a default optimization algorithm. In our proposed method, we have used the Adam optimizer with a learning rate of 0.001. Learning rate is a key hyperparameter to set while training a neural network, because low learning rate results in a slow progress in training and tiny updates to network weights, whereas high learning rate has a risk of hastily converging to a suboptimal solution instead of the optimal solution. So we chose the recommended learning rate of Adam optimizer, which performed well on our dataset. In terms of loss function, for a multilabel classification using neural networks, the most commonly used loss functions are categorical cross entropy (CCE) and sparse categorical cross entropy (SCCE). CCE is used for cases where soft probabilities exist that allow a sample to have multiple probabilities (e.g. 0.6 probability of belonging to label x and 0.4 to label y). On the contrary, SCCE is used when output labels are mutually exclusive. As our case matches with the latter one, we have used sparse categorical cross entropy (SCEE) as the loss function. Parameters set for ANN classification are listed in Table 5.

Table 5. Parameters selected for ANN classification of alphabet dataset

Optimizer	**Adam**	Number of Dropout Layers	**2**
Learning Rate	**0.001**	Number of Hidden Layers	**4**
Loss Function	**SCEE**	Activation Function (Hidden Layers)	**ReLu**
Epochs	**100**	Activation Function (Output Layer)	**Softmax**

ANN Classification Results. As mentioned earlier, for the alphabet dataset, 80% data was used for training the model and 20% was stored for testing purposes. After 100 epochs, the model has yielded a 97.82% training accuracy, 99.55% validation accuracy and 99.41% testing accuracy, in 308s of computation time. Figure 10 shows classification results of ANN model on training and validation data :

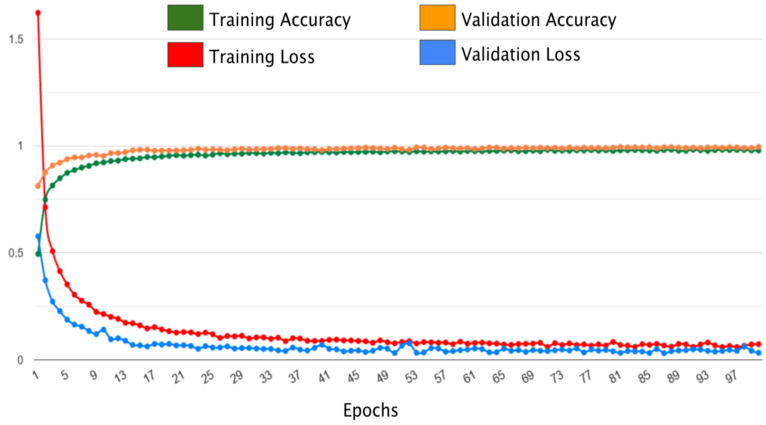


Fig. 10. Performance of ANN on training and validation data: alphabet dataset

Using the confusion matrix, we have evaluated accuracy for individual 37 signs. Figure 11 shows individual sign accuracy :

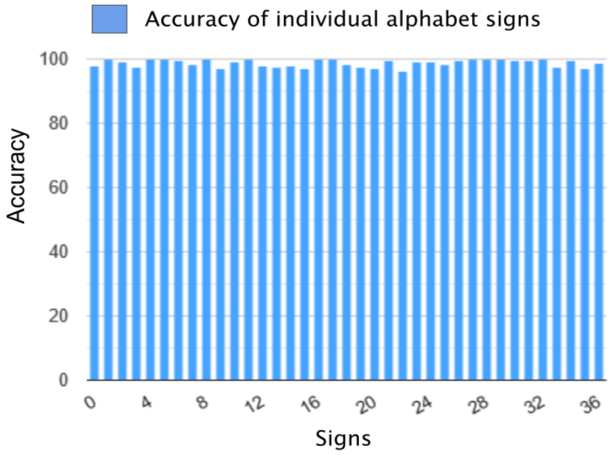


Fig. 11. Individual accuracy of 37 signs of alphabet dataset

Result Analysis. The ANN model performed very well on the alphabet dataset and yielded better classification accuracy compared to previous datasets. Since 63 coordinate values extracted from 21 hand key-points are entered as input features, and these values differ from sign to sign, the output labels were distinguishable from these features. That's why we can see a steep increase in the accuracy while training, from 54.7% to 91% within just 5 epochs (Fig. 10). As the computation time for an epoch was very low (3s on average), we continued upto 100 epochs.

Figure 11 shows the individual accuracy of the signs, which is very high for almost all the 37 sign letters. The reason for this near-perfect accuracy is the dissimilarity between the signs; due to the signs being different, the 63 coordinate values of one sign do not match that of the other. So, a sign could be clearly distinguishable from input features.

4.2 ANN Classification on the proposed BdSL47 dataset

In the second phase of our recognition, we have merged the alphabet dataset with the digits dataset, and named it BdSL47. The merged dataset consists of 47000 RGB image samples, and 470 corresponding csv files, containing x, y and depth values extracted from the samples. These csv files are the skeleton of the tabular dataset. This multilabel classification problem has 47 output labels. Like the alphabet dataset, 10 fold cross validation method was used here as well and the dataset was splitted into a train-test ratio of 80 : 20 prior to classification. On this dataset as well, we have initially performed classification using KNN, SVM and RFC, although it was estimated that the classification results will be similar to the alphabet dataset. Table 6 shows the results.

Table 6. Accuracy Results for KNN, SVM, RFC on the BdSL47 dataset

Classifier	Accuracy (%)
KNN	82.85
SVM	83.90
RFC	90.90

From Table 6, we can clearly see that the classification results are close to that of the alphabet dataset, which is not acceptable. Hence, for better classification accuracy, we have used same ANN architecture with some parameter changes. In this case, the model has an input layer of 63 nodes, an output layer of 47 nodes, and 4 hidden layers, with dropout rates of 25% in the last two hidden layers.

Optimizer, Learning Rate and Loss Function. Similar approach was used for setting the optimizer and loss function for this dataset as well; we have used Adam optimizer with a learning rate of 0.001 and sparse categorical cross entropy as the loss function.

ANN Classification Results. For the BdSL47 dataset also, the train test split ratio was 80 : 20. After 100 epochs, the training, validation and testing accuracy were respectively 96.36%, 97.54% and 97.78%, with a total computational time of 312 seconds. The accuracy is lower compared to the alphabet dataset, but it is still satisfactory. Using the confusion matrix, we have also evaluated accuracy for individual 47 signs. Fig. 12 and 13 respectively shows classification results and individual sign accuracy :

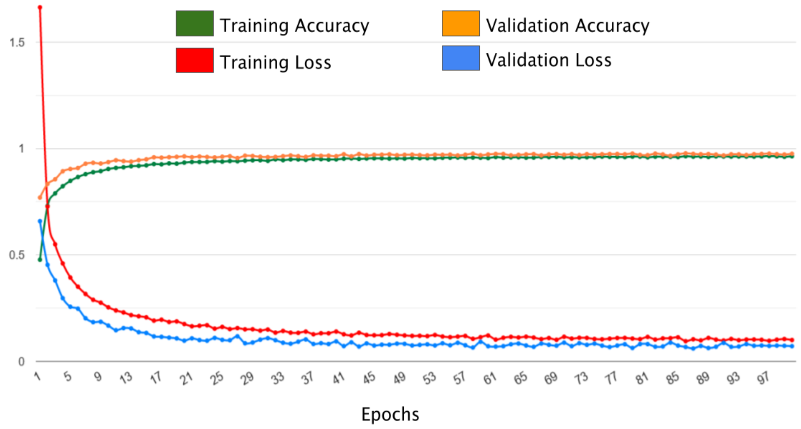


Fig. 12. Performance of ANN on training and validation data: BdSL47 dataset

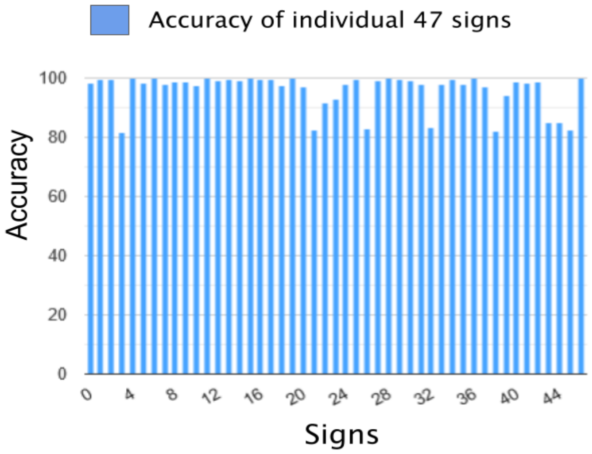


Fig. 13. Individual accuracy of 47 signs of BdSL47 dataset

Result Analysis. Like the previous case, the ANN model performed well on the BdSL47 dataset and yielded a decent accuracy. The dataset has 10 more output labels, however the model would show similar accuracy results, if all the signs were non-identical. But the fact is, some of the signs in the sign digits are almost identical to some of the signs in the sign alphabet, for example, sign 6 (৬) in sign digits is identical to sign 26 (ব) in sign alphabet. Because of their similarity, the coordinate values of these signs are very close; hence, distinguishing output labels from input features becomes more difficult.

From the individual accuracy bar graph (Figure 13), we can ascertain this reason. Accuracy of some of the letter-signs that were very high in the previous dataset, have been drastically decreased in the merged dataset; also accuracy of some of the digit-signs are almost as low as them. Figure 14 gives a visual representation of such cases :

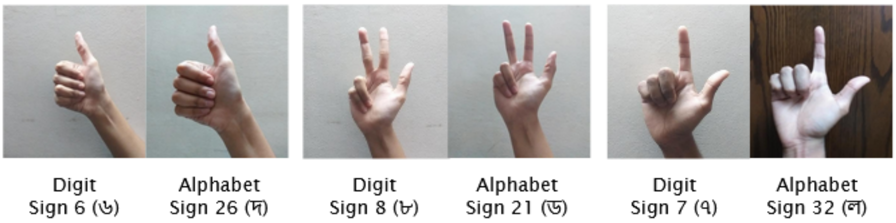


Fig. 14. Identical signs in sign-digits and sign-alphabet dataset

4.3 ANN Classification on the proposed BdSL47 dataset

As mentioned earlier, some research works have been conducted on Bangla Sign Language, and authors have built several datasets, mostly on alphabet. Table 7 shows a comparison among the datasets:

Table 7. Comparative analysis of structures of Bangla Sign Language datasets

Ref	Type	No. of Signs	Sign Type	Dataset Size (Images)	BG[2] and Lighting	Dataset Access
Rahaman et al, 2018	Digits Alphabet	46	Static 1-handed 2-handed	27600	Static	No
Rahaman et al, 2020	Digits Alphabet Others	52	Static 1-handed 2-handed	36400	Static	No

[2] BG = Background

AA[3], 2016	Alphabet	37	Static 1-handed	518	Static	No
Hoque et al., 2018	Alphabet	10	Static 2-handed	100	Random	Yes
Hoque et al., 2020	Alphabet	36	Static 2-handed	1200 (Augmented to 400K)	Random	Yes
Islam et al., 2018	Alphabet	36	Static 2-handed	1800	Static	Yes
Islam et al., 2018	Digits	10	Static 2-handed	1000	Static	Yes
Rayeed et al., 2022	Digits	10	Static 1-handed	10000 100 csv	Random	Yes
Ours	**Alphabet**	**37**	**Static 1-handed**	**37000 370 csv**	**Random**	**Yes**
Ours	**Digits Alphabet**	**47**	**Static 1-handed**	**47000 470 csv**	**Random**	**Yes**

Comparative Analysis. From Table 7, we can see that most of the BdSL datasets have been built based on two-handed static alphabet signs, and they were all image datasets, on which mostly deep learning approaches were performed for classification. On the contrary, in our case, although we have an image dataset, the foundation of BdSL47 is based on the 470 csv files, which form the tabular dataset.

[3] AA = Ahmed and Akhand, 2016

Because of this incompatibility, an equivalent comparison among the classification results of the datasets is not possible. However, we have listed the feature extraction techniques and classification models of the referenced datasets for a better overview in Table 8.

Table 8. Analysis of classification results on Bangla Sign Language datasets

Ref	**Type**	**No. of Signs**	**Feature Extraction Method**	**Classification Model**	**Accuracy (%)**
Rahaman et al, 2018	Digits Alphabet	46	FRB-RGB	Haar-KNN	95.67
Rahaman et al, 2020	Digits Alphabet Others	52	NOBV + WGV	NOBV + WGV	95.83
AA[4], 2016	Alphabet	37	BSL-FTP	ANN	98.99
Hoque et al., 2018	Alphabet	10	Faster R-CNN	CNN	98.20
Hoque et al., 2020	Alphabet	36	CNN	VGG19	99.10
Islam et al., 2018	Alphabet	36	CNN	CNN	94.74
Islam et al., 2018	Digits	10	CNN	CNN	92.87

[4] AA = Ahmed and Akhand, 2016

Rayeed et al., 2022	Digits	10	MediaPipe	SVM	98.65
Ours	**Alphabet**	**37**	**MediaPipe**	**ANN**	**99.41**
Ours	**Digits Alphabet**	**47**	**MediaPipe**	**ANN**	**97.78**

Result Analysis. From the comparison above, we can clearly see that the alphabet dataset has performed significantly better than other datasets. The merged dataset, BdSL47 has also performed very well, considering the homogeneity of some signs in digits dataset with some signs in alphabet dataset. The classification process has taken less computation time compared to others, because our model has been run on a tabular dataset, unlike most of the other BdSL datasets.

4.4 Comparison with the reference dataset Ishara Lipi

As we have considered Ishara Lipi as our reference dataset, we delineated a comparative analysis between the two datasets. Ishara Lipi contains two-handed static signs of the 36 Bangla sign alphabet, with 50 RGB image samples per sign character. Prior to classification, the samples were cropped, resized to 128×128 and converted to Grayscale;

and the samples were taken in front of a white background. For getting an exact comparative overview, we have also taken 50 samples per sign, and only the samples with a white background. Moreover, we have resized the samples, and converted them to Grayscale.

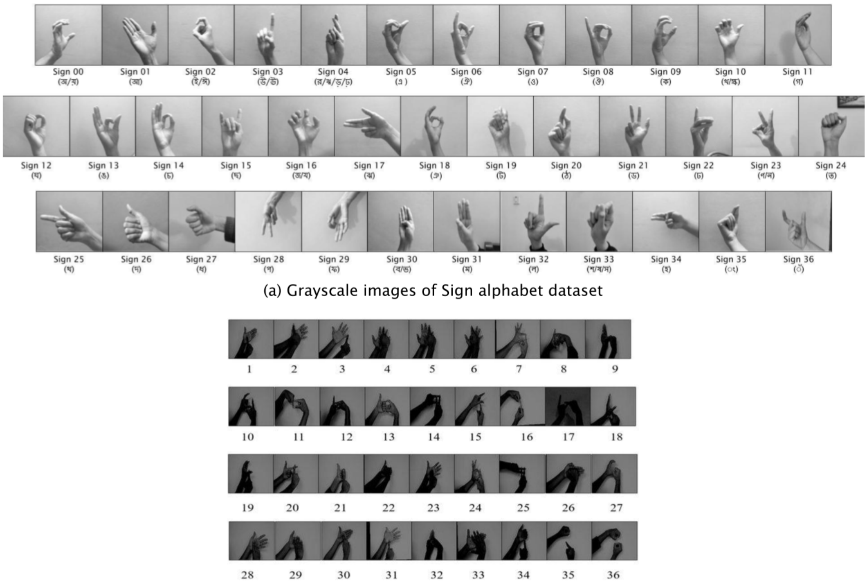
For classification in the Ishara Lipi dataset, the authors have proposed a 9-layer CNN architecture, with two sets of Convolutional (with kernel size 32 and 64) and Max Pooling Layer. Adam optimizer was used for optimization, with a learning rate of 0.001%. We have adopted their CNN architecture, with some hyperparameter tuning. On our dataset, the model yielded a better accuracy with a lesser number of epochs. For a better overview, some important factors of comparing the two datasets are stated in Table 9.

Table 9. Accuracy Results for KNN, SVM, RFC on the BdSL47 dataset

Factors	**Ishara Lipi dataset**	**Sign alphabet dataset**
No. of Samples	36	37
Samples per sign	50	50
Total Samples	1800	1850
Sample Type	Two-handed, Static	One-handed, Static
Background	White	White

Image Size	128×128	128×128
Image Type	RGB (Converted to Grayscale)	RGB (Converted to Grayscale)
Optimizer	Adam	Adam
Accuracy	94.74%	98.92%

Result Analysis. The classification results show that our dataset outperformed the Ishara Lipi dataset using the same CNN architecture, despite being homogeneous in the structure and size. Because of signs being two-handed in the former dataset, these gestures were more complex for recognition. On top of that, after resizing into 128×128 and converting to grayscale, most probably due to lower lighting ambience, some parts of several signs became too dark in the Ishara Lipi dataset. Compared to that, after resizing and converting to grayscale, the signs are more clearly visible and easily distinguishable in our dataset, which resulted in a significant increase in classification accuracy. Figure 15 shows a comparison between grayscale images of the datasets.



(a) Grayscale images of Sign alphabet dataset

(b) Grayscale images of Ishara Lipi dataset

Fig. 15. Comparison between grayscale images of (a) sign alphabet dataset and (b) the reference Ishara Lipi dataset

5 Conclusion and Future Work

We believe that the proposed open-access dataset, BdSL47, will be a great resource for future research works in Bangla sign language recognition. The framework used for extracting depth information, MediaPipe, has been a cost-efficient way of constructing a depth-based dataset that outperformed the existing sign alphabet datasets, with lesser computation time. However, there are some limitations and drawbacks that we found while working on MediaPipe. In brighter lighting ambience and plain background, the hand key-point detection is more accurate, whereas in darker lighting ambience and complex background, there were some faulty detections. The proposed dataset has some limitations too. In order to impose variations in the dataset, we used scaling, translation, rotation and other factors. However, due to controlled settings, the variations were not completely natural, therefore, the dataset still lacks challenging data. In future, with dynamic setups, we intend to make the dataset more challenging. Furthermore, from the extracted coordinate values, we can conduct feature engineering and use other features, such as finger-foldedness, finger-height, angles between fingers for classification purpose, unlike this case, where the normalized coordinate values extracted from MediaPipe have been directly used as the input features.

From the classification results above, we can say that both neural network models have performed well on the proposed dataset BdSL47. Our next goal is to apply multimodal classification on the proposed dataset. Additionally, the proposed dataset only contains one-handed static signs; we would like to extend our research on dynamic signs as well. Apart from research works, a real-time Bangla sign recognition system can be implemented based on depth information.

Compliance with Ethical Standards

1. **Funding:**

   This research study has been funded by Institute for Advanced Research (IAR), United International University (UIU), Dhaka, Bangladesh. (Grant Reference : UIU-IAR-01-2022-SE-37)

2. **Conflict of Interest:**

   This research study has no conflicts of interest to disclose.

3. **Ethical Conduct:**

   We confirm that this research work is original and has not been published elsewhere, nor is it currently under consideration for publication elsewhere.

4. **Data Availability Statements:**

   The proposed dataset generated and analyzed during this research study, is publicly available in Harvard Dataverse repository, https://doi.org/10.7910/DVN/EPIC3H

References

1. Rayeed, S.M., Akram, G.W., Tuba, S.T., Zilani, G.S., Mahmud, H. and Hasan, M.K., 2022, March. Bangla sign digits recognition using depth information. In Fourteenth International Conference on Machine Vision (ICMV 2021) (Vol. 12084, pp. 190-199). SPIE.
2. Islam, M.S., Mousumi, S.S.S., Jessan, N.A., Rabby, A.S.A. and Hossain, S.A., 2018, September. 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)* (pp. 1-4). IEEE.
3. Islam, M., Mousumi, S.S.S., Jessan, N.A., Rabby, A.K.M., Abujar, S. and Hossain, S.A., 2018, December. Ishara-Bochon: The First Multipurpose Open Access Dataset for Bangla Sign Language Isolated Digits. In International Conference on Recent Trends in Image Processing and Pattern Recognition (pp. 420-428). Springer, Singapore.
4. Hoque, O.B., Jubair, M.I., Akash, A.F. and Islam, S., 2020. Bdsl36: A dataset for bangladeshi sign letters recognition. In Proceedings of the Asian Conference on Computer Vision.

5. Rahaman, M.A., Jasim, M., Ali, M.H. and Hasanuzzaman, M., 2015, December. Computer vision based bengali sign words recognition using contour analysis. In 2015 18th International Conference on Computer and Information Technology (ICCIT) (pp. 335-340). IEEE.
6. Rahaman, M.A., Jasim, M., Ali, M.H., Zhang, T. and Md. Hasanuzzaman, 2018. A real-time hand-signs segmentation and classification system using fuzzy rule based RGB model and grid-pattern analysis. Frontiers Comput. Sci., 12(6), pp.1258-1260.
7. Rahaman, M.A., Jasim, M., Ali, M. and Hasanuzzaman, M., 2020. Bangla language modeling algorithm for automatic recognition of hand-sign-spelled Bangla sign language. Frontiers of Computer Science, 14(3), pp.1-20.
8. Ahmed, S.T. and Akhand, M.A.H., 2016, December. Bangladeshi sign language recognition using fingertip position. In 2016 International conference on medical engineering, health informatics and technology (MediTec) (pp. 1-5). IEEE.
9. Hoque, O.B., Jubair, M.I., Islam, M.S., Akash, A.F. and Paulson, A.S., 2018, December. Real time bangladeshi sign language detection using faster r-cnn. In 2018 international conference on innovation in engineering and technology (ICIET) (pp. 1-6). IEEE.

10. Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.L. and Grundmann, M., 2020. Mediapipe hands: On-device real-time hand tracking. arXiv preprint arXiv:2006.10214.
11. Halder, A. and Tayade, A., 2021. Real-time vernacular sign language recognition using mediapipe and machine learning. Journal homepage: www. ijrpr. com ISSN, 2582, p.7421.
12. Domènech López, A., 2020. Sing Language Recognition-ASL Recognition with MediaPipe and Recurrent Neural Networks (Bachelor's thesis, Universitat Politècnica de Catalunya).
13. Ghosh, S. (2021). Proposal of a Real-time American Sign Language Detector using MediaPipe and Recurrent Neural Network. *International Journal of Computer Sciences and Engineering*, 9(7), pp.46–52. doi:10.26438/ijcse/v9i7.4652.
14. Harris, M. and Agoes, A.S., 2021, November. Applying Hand Gesture Recognition for User Guide Application Using MediaPipe. In *2nd International Seminar of Science and Applied Technology (ISSAT 2021)* (pp. 101-108). Atlantis Press.
15. Herath, R.J. and Ishanka, P., 2022. An Approach to Sri Lankan Sign Language Recognition Using Deep Learning with MediaPipe. In International Conference on Digital Technologies and Applications (pp. 449-459). Springer, Cham.

16. Rayeed, S. M. (2022). *BdSL47: A complete dataset of sign alphabet and digits of Bangla Sign Language (BdSL) using depth information via MediaPipe* (Version V1) [Computer software]. Harvard Dataverse. https://doi.org/10.7910/DVN/EPIC3H
17. Doliotis, P., Stefan, A., McMurrough, C., Eckhard, D. and Athitsos, V., 2011, May. Comparing gesture recognition accuracy using color and depth information. In *Proceedings of the 4th international conference on PErvasive technologies related to assistive environments* (pp. 1-7).
18. Pugeault, N. and Bowden, R., 2011, November. Spelling it out: Real-time ASL fingerspelling recognition. In *2011 IEEE International conference on computer vision workshops (ICCV workshops)* (pp. 1114-1119). IEEE.
19. Rodriguez, K.O. and Chavez, G.C., 2013, August. Finger spelling recognition from RGB-D information using kernel descriptor. In *2013 XXVI Conference on Graphics, Patterns and Images* (pp. 1-7). IEEE.
20. Liang, H. and Yuan, J., 2014. Hand parsing and gesture recognition with a commodity depth camera. In *Computer Vision and Machine Learning with RGB-D Sensors* (pp. 239-265). Springer, Cham.
21. Mudduluru, S., 2017. Indian Sign Language Numbers Recognition using Intel RealSense Camera.

22. Almasre, M.A. and Al-Nuaim, H., 2016. A real-time letter recognition model for Arabic sign language using kinect and leap motion controller v2. *International Journal of Advanced Engineering, Management and Science*, *2*(5), p.239469.

23. Lang, S., Block, M. and Rojas, R., 2012, April. Sign language recognition using kinect. In International Conference on Artificial Intelligence and Soft Computing (pp. 394-402). Springer, Berlin, Heidelberg.

24. Liao, B., Li, J., Ju, Z. and Ouyang, G., 2018, June. Hand gesture recognition with generalized hough transform and DC-CNN using realsense. In *2018 Eighth International Conference on Information Science and Technology (ICIST)* (pp. 84-90). IEEE.

25. Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

26. Al Marouf, A., Hasan, M.K. and Mahmud, H., Fingertip Detection and Finger Identification Approach for Hand Gesture Recognition using Microsoft Kinect.

27. Mahmud, H., Hasan, M., Kabir, M. and Mottalib, M.A., 2018. Recognition of symbolic gestures using depth information. *Advances in Human-Computer Interaction*, *2018*