Bangla Sign Digits Recognition using Depth Information

S. M. Rayeed[1], Gazi Wasif Akram[1], Sidratul Tamzida Tuba[1], Golam Sadman Zilani[1], Hasan Mahmud[1], Md. Kamrul Hasan[1]

[1]Systems and Software Lab, Department of Computer Science and Engineering, Islamic University of Technology (IUT), Gazipur, Bangladesh

ABSTRACT

Sign Language Recognition (SLR) targets on interpreting the sign language into text or speech, so as to facilitate the communication between deaf-mute people and ordinary people. The task has broad social impact, but is still very challenging due to the complexity and large variations in hand actions. Existing dataset for Sign Language Recognition (SLR) in Bangla Sign Language (BdSL) is based on RGB images. Recent research on sign language recognition has shown better recognition accuracy using depth-based features. In this paper, we present a complete dataset for Bangla sign digits from Zero (Shunno in Bangla) to Nine (Noy in Bangla) using MediaPipe, a cross-platform depth-map estimation framework. The proposed method can utilize hand skeleton joint points containing depth information in addition to x, y coordinates from RGB images only. To validate the effectiveness of our proposed approach, we have run MediaPipe on a benchmark American Sign Language (ASL) dataset. Running different classifiers in our proposed dataset we got 98.65% using Support Vector Machine (SVM). Moreover, we compared our dataset with the existing Bangla digit dataset Ishara Bochon using deep learning based approach and achieved significantly higher accuracy.

Keywords: Bangla Sign Language (BdSL), Hand Gesture Recognition, Depth Information, MediaPipe, Palm Detection Model, Hand Landmark Model, Hand Key-points.

1. INTRODUCTION

Sign language is a kind of visual language that uses hand shapes, facial expression, gestures and body language. Hearing-impaired people use sign language to communicate with others. It is the basic method of communication for them, but it is difficult to interpret for most of the people, so the communication needs to be developed, and that's where machine learning may become handy. Pre-trained machine translators can be helpful to ease the interpretation of what a person is trying to express, therefore easing the communication.

From the technological aspect, with the advancement in technology, the ways of user interaction have been changing from physically operating devices via keyboard to, interacting without any touch such as gestures, virtual reality etc. Gesture recognition has been one of the main focuses in recent times in the field of computer science in various areas. 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. Sign language can be a collection of specified hand gestures, so several computer vision based solutions applied in gesture recognition are becoming well-known for sign language recognition all over the world. In Bangladesh, almost 2.6 million people are deaf and mute. To help them communicate with the rest of our community, in 1974, a book named Bengali Sign Language Dictionary was published by the National Centre for Special Education Ministry. Another book named Ishara Bhasay Jogajog was reprinted in 2015 for the learning purpose of deaf and mute children in Bangladesh. In terms of recognition in Bangla Sign Language, it has got off to a start, when some of the Deep Learning enthusiasts Sanzid et. al. published a dataset for sign digits [1] and another for sign letters [2] in Bangla Sign Language (BdSL). In both cases, RGB images were used for dataset generation, and Convolutional Neural Network (CNN) was used for classification.

Recent research works on other sign languages and hand gesture recognition have shown that, instead of just relying on RGB images, extracting depth information from a given hand shape and using it for classification can yield a better accuracy. For extracting the depth information, there are some devices that use 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. But such devices are very costly. As a low-cost alternative, MediaPipe framework is very efficient. This framework can depict a geometric representation of the hand skeleton from RGB images taken by webcam or mobile, detect the coordinates of 21 key-points from the hand skeleton, estimate a depth value of these points using depth map, with no extra devices. Although in this case, the depth values are estimated, as opposed to accurate sensor-provided depth values from aforementioned devices. But still these estimated values can effectively be used as spatial features and provide higher accuracy, as shown in [3], where multiple sign language datasets such as American, Indian, Italian and Turkey had been used for training to analyze the capability of MediaPipe, where average accuracy was 99%.

So as a cost-effective way of using depth information, in this paper, we have presented a dataset of static hand gestures of sign digits in Bangla Sign Language (BdSL) using depth information via MediaPipe, that detects 21 hand key-points, measures the x and y coordinate values and estimates the depth values of these points. The coordinate values were normalized and stored. Our dataset consists of these values, detected from RGB image samples. Then we have classified our dataset using different classifiers, such as SVM, KNN, XGB. Also for comparing our dataset with Ishara Bochon, the existing dataset in Bangla Sign Language (BdSL), we have used our RGB image samples as an image dataset, and applied the same CNN architecture as Ishara Bochon on that dataset, which yielded a higher accuracy than Ishara Bochon. Moreover, to validate the efficiency of the proposed MediaPipe framework, we have conducted an experiment on a benchmark dataset of American Sign Language [4] that contains 10 static signs from digit 0 to 9 and 24 static signs from A to Z (except J and Z). We have applied the MediaPipe framework on this benchmark ASL dataset as well to extract depth information and evaluate the performance on different classification models.

2. LITERATURE REVIEW

Sign language involves non-vocal communication with a combination of hand gestures, lip patterns and facial expressions. In this paper, we are focused only on the static hand gestures. In the early 16th century, an Italian physician Geronimo Cardano, declared that it is necessary to take care of the deaf community and they should be taught how to communicate with the world. Juan Pablo de Bonet created and published the first book on sign language in 1620. In 1755, Abbe Charles-Michel delEpee created the first sign language school with no cost to students in France. Later, he created finger spelling and gestures to recognize phrases and words. Later on, different sign languages have been introduced for deaf and mute communities all over the world.

2.1 Ishara Bochon: Existing Dataset for Digits in Bangla Sign Language

In 2019, Islam et. al. published the first multi purpose open access dataset for sign digits in Bangla Sign Language, Ishara Bochon. The dataset contains 1000 RGB image samples of 10 bangla sign digits (0, 1, 2, .., 8, 9), collected from deaf-mute communities and general volunteers. After collecting raw RGB images, data preprocessing was performed for recognition purposes. The dataset generation of Ishara Bochon is divided into following states -

1. Capturing images: The dataset contains 1000 images of sign digits. The authors have collected the RGB image samples of uncovered hands in white background, from many deaf-mute school communities.
2. Labeling data: The next step was to categorize the image samples into 10 categories, one for each sign (from 0 to 9). After labeling, there were 100 RGB samples in each category.
3. Cropping images: Cropping is necessary for better feature extraction from the image samples, specially for CNN classification. So the images were cropped to show the region of interest.
4. Resizing image and converting to grayscale: For easier and more accurate classification in deep learning and computer vision based works, images were resized into 128×128 pixels, then converted from RGB to grayscale.

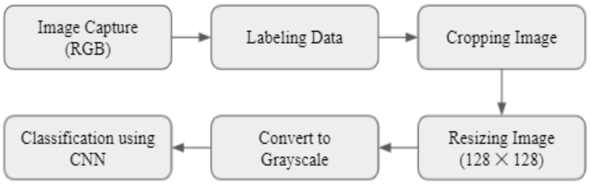

Figure 1. Overview of Ishara Bochon

After data preprocessing, the images were fitted into a 8-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 yielded 92.87% validation accuracy and 96.52% training accuracy.

2.2 Depth Information in Sign Language Recognition (SLR)

Sign language recognition was first introduced using the RGB images. Later, several researches on gesture recognition have stated that using depth information gives an extra advantage (information about z-coordinate) for classification, hence yields a better accuracy. While comparing the accuracy in hand gesture recognition between depth and RGB images, Doliotis et. al. have shown that depth images outperform RGB images in hand gesture recognition, specially in case of complex backgrounds [5]. In their experiment with Microsoft Kinect depth sensing camera, they firstly used easy data - static hand gestures having a simple and stable background, then hard data - static hand gestures with people moving in the background. Results were in favour of depth information in both cases.

There are several other research works on other sign languages using depth information, that delineates how the performance gets improved by using depth information. Rodriguez et al. proposed a hand gesture recognition model on American Sign Language (ASL) dataset, by hand segmentation using depth map [7]. Firstly, the hand area was segmented and precise hand shapes were extracted for gesture recognition. Then, features were extracted using Scale-Invariant Feature Transformation (SIFT) from image samples. Finally, these features were used as the input for recognition, where SVM yielded the highest accuracy of 90.2%. Liang et al. proposed a distance-adaptive feature selection method for extracting more discriminative depth-context features for hand recognition [9], and the experiment yielded 17.2% higher accuracy on the synthesized dataset for single-frame parsing.

In later research works, depth sensing devices became popular for extracting depth information. In Indian Sign Language (ISL) digit recognition, Intel RealSense was used to extract depth information of 22 hand-joint points from sign digits [10]. Then several classifiers were used for classification, among which SVM performed best, with an accuracy of 93.5%. Al-Nuaim et. al. proposed a supervised machine learning model for hand gesture recognition using Microsoft's Kinect with a Leap Motion Controller, on the 28 letters of Arabic Sign Language [11]. Their proposed model relies on predicting the hand pose from two depth images and then defining a classifier algorithm, based on 3D positions of hand-joints of the corresponding letters in real time. The proposed ArSL model recognized 22 of the 28 Arabic alphabets with an accuracy of 100%. Lang et. al. presented an open source framework for hand gesture recognition using Microsoft Kinect, with isolated 25 signs of German Sign Language (GSL). The framework is robust and efficient for new gestures, as it requires performing new gestures several times in front of the camera. Recognition is done using HMM (Hidden Markov Model), with an accuracy of 97% [12]. In [13], Mistry et. al. have compared the classifiers - Support Vector Machine (SVM) and Artificial Neural Network (ANN), for recognizing American Sign Language (ASL) using Intel RealSense. The dataset they used consisted of normalized x, y and z coordinate values of hand key-points, extracted from 26000 images of 10 users. In result analysis, SVM yielded an accuracy of 95%, outperforming ANN's accuracy of 92.1%. In 2018, Liao et. al. proposed a hand gesture recognition model based on generalized Hough transform on English alphabet, using Intel RealSense, by mapping depth images to color images for hand segmentation in complex background. For classification, a Double-Channel Convolutional Neural Network (DC-CNN) model was proposed, with dedicated channels for RGB and depth images. The proposed model yielded an accuracy of 99.4% [14].

Research works on other sign languages using the proposed framework, MediaPipe : In case of most of the aforementioned research works, depth sensing devices have been used,which are quite expensive. MediaPipe, the proposed framework is a low-cost alternative of extracting depth information from RGB images. This is easier, more comfortable and affordable, as it requires no additional sensors. Although the framework has been introduced recently, there has been some research on other sign languages using MediaPipe. In 2021, Halder et. al. have used multiple sign language datasets such as American, Indian, Italian and Turkey for training purposes to analyze the capability of the MediaPipe open-source framework. They proposed a lightweight predictive model that is adaptable to smart devices. With an average accuracy of 99%, their proposed model was efficient, precise and robust [3]. In 2020, Antonio López used Google's MediaPipe framework to recognize four different gestures of American Sign Language (Hello, No, Sign and Understand). Recurrent Neural Networks (RNN) was used for classification, which yielded an accuracy of 92% [15].

Apart from hand-signs recognition, in general, character recognition in texts and signatures is a widely-researched topic. In [30], Wickramaarachchi et. al. proposed an algorithm that extracts scale and non-scale variant features from handwritten signatures, and used SVM for classification. In [6], Ceniza et. al. proposed a mobile application that can recognize text in degraded images using Adaptive Document Image Binarization and achieved 93.17% accuracy in individual character recognition. Liu et. al. proposed a CNN architecture for Oracle Bone Inscriptions Recognition, where collected images of oracle-like characters (e.g. Ren, Zi) were given as input [8]. The model extracts image features and implicitly learns from training data to recognize oracle-like characters, showing an accuracy of 94.2%.

3. MEDIAPIPE FRAMEWORK FOR DEPTH INFORMATION

MediaPipe is a framework for building multimodal (e.g. video, audio), cross platform (i.e Android, iOS devices) applied ML pipelines. With MediaPipe, a perception pipeline can be built as a graph of modular components, including, for instance, inference models (e.g. TensorFlow) and media processing functions. Cutting edge machine learning models using MediaPipe include – face detection, multi-hand tracking, hair segmentation, 3D object detection and tracking etc.

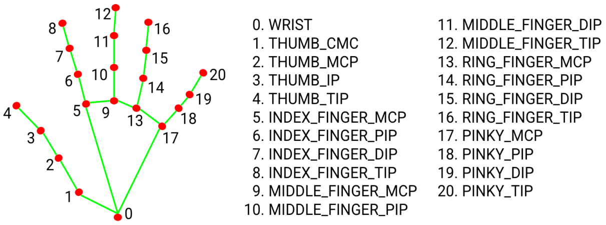

Figure 2. MediaPipe Hand Landmarks – 21 Hand Key-points

The module used here is MediaPipe Hands, consisting of two models – Palm Detection Model and Hand Landmark Model. This is a high-fidelity solution for hand and finger tracking. Hand landmark model operates on the cropped image region defined by the Palm Detection Model and returns high-fidelity 3D hand key-points.

Palm Detection Model, BlazePalm: To detect initial hand locations, a single-shot detector model, optimized for mobile real-time application was designed, named BlazePalm [27] similar to BlazeFace [26]. BlazePalm is designed to work across a variety of hand sizes with a large scale span and is capable of detecting occluded and self-occluded hands. First, a palm detector was trained instead of a hand detector, since estimating bounding boxes of rigid objects like palms and fists is significantly simpler than detecting hands with articulated fingers. Second, an encoder-decoder feature extractor, similar to Feature Pyramid Networks [28], was used for a larger scene-count, even for small objects.

Hand Landmark Model: After running BlazePalm model over the whole image sample, subsequent hand landmark model performs precise landmark localization of 21 3D hand-knuckle coordinates inside the detected hand regions via regression, that is direct coordinate prediction. The model learns a consistent internal hand pose representation and is robust even to partially visible hands and self-occlusions. The hand landmark model has three outputs - (i) a hand flag indicating the probability of hand presence in the input image, (ii) 21 hand landmarks consisting of x, y, and relative depth (z) values, and (iii) a binary classification of handedness, e.g. left or right hand. (See Figure 3)

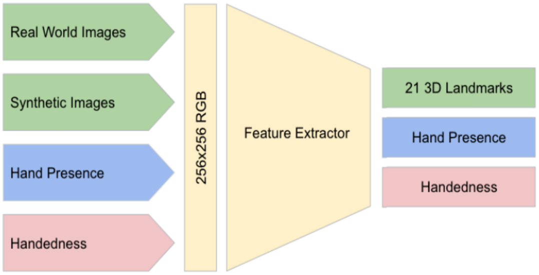

Figure 3. MediaPipe Hand Landmark Model Architecture [27]

For hand key-points detection, a similar work has been done by Simon et. al. that uses Multi-view Bootstrapping for the 21 landmarks [29]. In the hand landmark model, the x and y coordinates are learned from both real-world images as well as synthetic datasets, with the relative depth with respect to the wrist point being learned only from synthetic images. A synthetic image is obtained by pure computation, i.e. by modelling real world images and simulating the laws of optics. The synthetic hand model was rigged with 24 bones and included 36 blendshapes, which control fingers and palm thickness. To obtain ground truth data, 30K real-world images were manually annotated with 21 3D coordinates. Binary classification was developed to predict whether the input hand is left or right.

4. PROPOSED APPROACH

This section contains the steps of generating the dataset using the MediaPipe framework. The dataset consists of 10,000 RGB images performed by 10 users with 10 signs each. Statistical information of the dataset is given below -

Table 1. Dataset statistics of our proposed dataset

Number of signs	10 (from 0 to 9)	Size of image samples	640× 480
Number of users	10	Number of input features	63
Number of input samples	10× 10×100 = 10000	Number of output labels	10

To build the dataset from scratch, we had to go through some states. Our proposed approach involves –

4.1 Image Sample Collection

As stated above, the first step of building the dataset was to collect RGB image samples from real-time streaming. For every sign from a user, more than 200 frames were captured, from which only the best 100 were finalised. We took extra samples, because, in case of initial frames during streaming, sometimes the lighting ambience tends to be darker than usual. Also, due to movement of the user or the camera, some frames may get blurry and may cause faulty or no detection. Image samples were collected via general-purpose cameras, e.g. webcam. As only the user-hand was there in the field of view (fov) of the camera, no explicit hand segmentation method was used.

Variation in sample collection: While collecting the samples, we tried to consider variations as much as possible to make the dataset versatile and more challenging for classification. Variations can be categorized into 2 types –

1. **User variation:** A total number of 10 users have participated for the generation of our dataset. The variation in users occurred in terms of age, gender, shape of user-hand and skin-color of the user.
2. **Environment variation:** While taking samples from the users, we have considered environmental variation as well, such as, angle between camera-lens and the hand, lighting ambience, backgrounds , distance from camera to user-hand, different image orientations (landscape and portrait), different hand orientations facing the camera (front or back), user-hand in different positions of the image, e.g. top right, top-left.

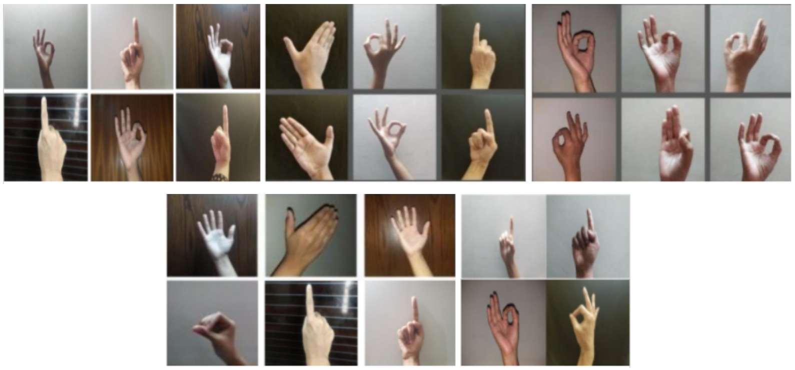

Figure 4. Dataset Variation (top row) background, hand-facing and angle ; (bottom row) luminance and users (age, hand-shape, sex)

4.2 Hand Key-points Detection and Pre-processing

After capturing the RGB images, the image frames were processed via MediaPipe for hand-tracking and hand key-points detection, by palm detection model and hand landmark model. This involves – detecting hand of the user using BlazePalm, detecting hand key-points using Hand Landmark Model, extracting normalized values of x and y coordinates, generating depth map from RGB images, extracting the estimated depth values for each hand key-point, and storing the normalized x and y values, along with estimated z values.

After that, we eliminated some samples where faulty or no detection occurred. Such cases happened when the user had to initially adjust his/her hand position and some frames got blurry due to movement of the user or the camera. We manually checked the detection output of each sample and selected the best 100 ones for a final run.

4.3 CSV Files Generation

After manual checking and elimination, we selected 100 frames that were the most-accurate ones. After a final processing of these images via MediaPipe, a corresponding .csv file was created for each sign performed by a user, which consists of the normalized x, y and z coordinate values of the 21 hand key-points, having 65 columns and 101 rows (with sample name and labeling). The complete dataset is available here upon request.

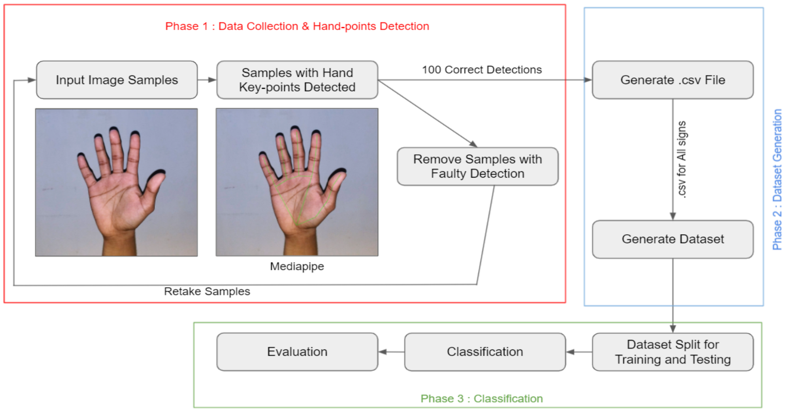
Phase 1 : Data Collection & Hand-points Detection
Input Image Samples
Samples with Hand Key-points Detected
100 Correct Detections
Remove Samples with Faulty Detection
Mediapipe
Retake Samples
Generate .csv File
.csv for All signs
Generate Dataset
Phase 2 : Dataset Generation
Dataset Split for Training and Testing
Classification
Evaluation
Phase 3 : Classification

Figure 5. Overview of the proposed approach

5. CLASSIFICATION

For this research, we have conducted classification on three different datasets and analysed the results. Firstly, we used different classifiers, such as KNN, SVM etc. on our proposed csv dataset. Then, we have performed CNN-based classification on our image dataset, with the CNN architecture being taken from existing Ishara Bochon recognition. Finally, for validating MediaPipe, we have applied it on the ASL dataset and evaluated the classification accuracy.

5.1 Classification on our Proposed Dataset

This is a 10-class classification problem, so we had 10 labels. Prior to classification, 10 fold cross validation had been used. The entire dataset was splitted into a train-test ratio of 0.80 : 0.20. As the coordinate values of 21 hand key-points were the input features, there were 21×3, or 63 input features. For classification, we used several classifiers, such as SVM, KNN, RFC, DTC, XGB. The results achieved from these classifiers are as follows –

Table 2. Classification accuracy of different classifiers on our proposed dataset

Classification Model	Accuracy	Classification Model	Accuracy
KNN	94.60%	XGB	98.00%
SVM (Linear)	98.00%	RFC	97.75%
SVM (Polynomial)	93.35%	DTC	96.25%
SVM (RBF)	98.65%		

We have also calculated the classification results for each individual sign. Individual accuracy of predicting the signs digits from these classifiers are as follows –

Table 3. Classification accuracy of different classifiers on individual sign digits of our proposed dataset

Classification Model	**Sign 0 (%)**	**Sign 1 (%)**	**Sign 2 (%)**	**Sign 3 (%)**	**Sign 4 (%)**	**Sign 5 (%)**	**Sign 6 (%)**	**Sign 7 (%)**	**Sign 8 (%)**	**Sign 9 (%)**
KNN	98.09	95.91	99.50	99.47	97.57	97.99	99.50	96.48	99.03	100
SVM (Lin)	99.04	97.76	98.51	100	98.52	98.51	100	97.93	99.03	100
SVM (Poly)	99.03	97.75	95.69	97.42	93.43	94.95	99.50	98.96	97.09	100
SVM (RBF)	99.52	99.09	99.01	100	99.51	98.99	99.50	100	99.51	100
XGB	99.52	97.29	99.50	99.47	98.54	98.49	100	97.44	100	100
RFC	99.04	98.20	98.52	99.47	96.63	97.98	100	97.93	98.54	100
DTC	95.22	98.64	99.50	97.93	95.67	92.72	99.50	98.96	98.54	100

Result Analysis: Most of the classifiers have yielded a decent accuracy. As we have mentioned, it was a 10-class classification problem with 63 input features. As the 3D coordinate values were different for each sign, the classifiers gave higher accuracy. There were some incorrect classifications as well – mostly between 4 and 5, 1 and 7, 2 and 8. Because, these signs were similar to each other. In case of these signs, all other coordinate values are similar except three thumb key-points - thumb tip, thumb ip (interphalangeal), thumb mcp (meta carpo phalangeal). Among individual sign digits, sign 9 has been accurately predicted by all the classifiers. Other signs also yielded a decent accuracy. Among the classifiers, SVM (RBF) has performed best, with an average accuracy of 99.51%.

5.2 Comparison with Ishara Bochon

Besides using depth information for sign digits recognition, we have also done a ground-level comparison between our dataset and the existing dataset, Ishara Bochon, by replicating their proposed classification methods on our dataset. In Ishara Bochon, there are 1000 RGB images, of size 128×128, later converted to grayscale. So, first we had to resize the input samples into 128×128, then convert the images from RGB into grayscale.

In Ishara Bochon, they proposed an 8-layer CNN architecture for classification. Adam optimizer was used for optimization, with a learning rate of 0.001%. After 500 epochs, the model yielded a training accuracy of 96.52% and a validation accuracy of 92.87% [1]. We have used their proposed CNN architecture, with the same hyper-parameters. On our dataset, after 100 epochs, the CNN model yielded a training accuracy of 99.87% and a validation accuracy of 99.10%. We have also tested the performance of our dataset by tuning the hyper-parameters, such as changing kernel and pooling sizes, trying different optimizers with lesser number of epochs etc. In such cases as well, the model yielded decent accuracy, outperforming the accuracy of existing dataset. For a better view of comparison, some important factors of comparing the two datasets are stated as follows –

Table 4. Comparison between our proposed dataset and the existing dataset, Ishara Bochon

Factors	**Ishara Bochon**	**Proposed Dataset**
Samples	1000	10000
Background	White	Various
Image Size	128×128	128×128
Image Type	RGB (Grayscale)	RGB (Grayscale)
Epochs	500	i. 50, ii. 100
Optimizer	Adam	i. Adam, ii. SGD
Accuracy	92.87%	i. 97.70% (Adam, 50), ii. 99.10% (Adam, 100), iii. 99.30% (SGD, 100)

Result Analysis: As the comparison table shows, the CNN model yielded highest accuracy on the proposed dataset with stochastic gradient descent (SGD) optimizer, which is 99.30%. Although in other cases as well, classification results on the proposed dataset are better than the existing dataset, and that with a lesser number of epochs. Moreover, compared to the existing dataset, the proposed one is larger in size, more versatile and challenging for recognition, with a number of user and environment variations as aforementioned.

5.3 Classification on benchmark American Sign Language (ASL) Dataset

As mentioned above, to validate the effectiveness of the proposed MediaPipe framework, we have tested this framework on a benchmark American Sign Language Dataset. The dataset was published in 2011 by Barczak et. al., who proposed a static hand gesture dataset for gestures in American Sign Language (ASL) by hand segmentation, noise reduction and feature extraction from input RGB images [4]. Statistical information of the ASL dataset is given below -

Table 5. Dataset statistics of benchmark American Sign Language (ASL) dataset

Number of input images	2524	Image type	RGB
Number of digit signs	10 (from 0 to 9)	Sign/Hand-gesture type	Static
Number of letter signs	24 (from A to Z, excluding J and Z)	Background	Black

After data augmentation and preprocessing the image samples of 10 sign digits, each of size 400 × 400, we stored the x, y, z coordinate values from the images and used different classifiers for recognition, and results are as follow –

Table 6. Classification accuracy of different classifiers on sign digits of American Sign Language (ASL) dataset

Classification Model	**Accuracy**	**Classification Model**	**Accuracy**
KNN	97.70%	XGB	99.70%
SVM (Linear)	100.00%	RFC	98.10%
SVM (Polynomial)	98.70%	DTC	96.80%
SVM (RBF)	99.90%	ANN	97.70%

5.4 Comparison with other research works on the same ASL Dataset

As we have mentioned above, several research studies have been done based on this benchmark ASL dataset proposing various methods. The following table shows a comparison among them –

Table 7. Comparison among different approaches and their achieved results on American Sign Language (ASL) dataset

Ref. No.	ASL Dataset	Feature Extraction Method	Classification Model	Accuracy (%)
[16]	A - J	Discriminative Zernike Moments	KNN	98.51
[17]	Digits	CNN	CNN	97.00
[17]	Letters	CNN	CNN	82.50
[18]	Digits	CNN	CNN	100
[19]	Letters	Spatial Pyramid Pooling	CNN	99.64
[21]	Letters	CNN	CNN	94.34
[23]	Both	YOLOv3 and VGG16	CNN	97.41
[22]	Letters	Inceptionv3	CNN	98.81
[20]	Digits	Gabor-Filter	CNN	99.90
[20]	Letters	Gabor-Filter	CNN	99.06
[24]	Both	AlexNet	SVM	99.82
[24]	Both	VGG16	SVM	99.76
[25]	Letters	Histogram of Gradient	SVM	98.67
[25]	Letters	HOG-LBP (Local Binary Pattern)	SVM	99.22
[25]	Letters	Generalized Search Tree	SVM	99.03
Ours	**Digits**	**MediaPipe**	**SVM**	**100**

Result Analysis: From the table we can see that MediaPipe yields a very decent accuracy on the sign digits, compared to others. Although in the case of [18], the proposed CNN architecture yielded an accuracy of 100.00% on ASL digit dataset, the processing time was about 75 seconds (74.7466 seconds). Whereas using MediaPipe, we have achieved the same accuracy with SVM, which takes less computational time.

6. CONCLUSION AND FUTURE WORK

From the classification results, we can say that our dataset performed well. As a cost-efficient way of using depth information in sign language recognition, MediaPipe has outperformed the existing dataset. Still there are some limitations and drawbacks that we found while working on MediaPipe. In darker lighting ambience and complex background, some faulty detection occurred. However, with the depth information via MediaPipe, classification was quicker, easier and better than the existing one.

The dataset is for sign digits in Bangla Sign Language. In future, we can replicate the process and develop another dataset for sign letters. Also, we can work for the betterment of this dataset by taking more samples from more users in different environmental setups, with as many variations as possible, to make the dataset more versatile and challenging. Besides, we can do feature engineering and consider other features instead of 3D coordinate values, such as finger-foldedness, finger-height, angles between fingers and evaluate the performance. This dataset consists of single-handed signs only, we can also start building a dataset using two-handed signs. Moreover, our work is focused on only static signs, we can further work on dynamic hand gesture recognition.

REFERENCES

[1] M.S. Islam, S.S.S. Mousumi, N.A. Jessan, A.S.A. Rabby and S.A Hossain (2019). Ishara Bochon: The First Multi-purpose Open Access Dataset for Bangla Sign Language Isolated Digits. 420–428. https://doi.org/10.1007/978-981-13-9181-1_37

[2] M.S. Islam, S.S.S. Mousumi, N.A. Jessan, A.S.A. Rabby and S.A Hossain (2018). Ishara Lipi: The First Complete Multipurpose Open Access Dataset of Isolated Characters for Bangla Sign Language. In 2018, International Conference on Bangla Speech and Language Processing (ICBSLP). IEEE, 1–4.

[3] A. Halder and A. Tayade (2021). Real-time Vernacular Sign Language Recognition using MediaPipe and Machine Learning. Journal homepage: https://www.ijrpr.com ISSN 2582 ([n. d.]), 7421.

[4] A.L.C. Barczak, N.H. Reyes, M. Abastillas, A. Piccio and T. Susnjak (2011). A New 2D Static Hand Gesture Colour Image Dataset for ASL Gestures. Research Letters in the Information and Mathematical Sciences 15 (2011), 12–20.http://www.massey.ac.nz/massey/fms/Colleges/College%20of%20Sciences/IIMS/RLIMS/Volume%2015/GestureDatasetRLIMS2011.pdf

[5] P. Doliotis, A. Stefan, C. McMurrough, D. Eckhard and V. Athitsos (2011). Comparing gesture recognition accuracy using color and depth information. In Proceedings of the 4th international conference on Pervasive technologies related to assistive environments. 1–7.

[6] A.M. Ceniza, T.K.B. Archival and K.V. Bongo, "Mobile Application for Recognizing Text in Degraded Document Images Using Optical Character Recognition with Adaptive Document Image Binarization," Journal of Image and Graphics, Vol. 6, No. 1, pp. 44-47, June 2018. doi: 10.18178/joig.6.1.44-47

[7] K.O. Rodriguez and G.C. Chavez (2013). Finger spelling recognition from RGB-D information using kernel descriptor. In 2013 XXVI Conference on Graphics, Patterns and Images. IEEE, 1–7.

[8] M. Liu, G. Liu, Y. Liu and Q. Jiao, "Oracle Bone Inscriptions Recognition Based on Deep Convolutional Neural Network," Journal of Image and Graphics, Vol. 8, No. 4, pp. 114-119, December 2020.

doi: 10.18178/joig.8.4.114-119

[9] H. Liang and J. Yuan (2014). Hand parsing and gesture recognition with a commodity depth camera. In Computer Vision and Machine Learning with RGB-D Sensors. Springer, 239–265.

[10] S. Mudduluru (2017). Indian Sign Language Numbers Recognition using Intel RealSense Camera. (2017).

[11] H. Al-Nuaim and M.A. Almasre (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 (2016).

[12] S. Lang, M. Block and R. Rojas (2012). Sign language recognition using kinect. In International Conference on Artificial Intelligence and Soft Computing. Springer, 394–402.

[13] J. Mistry and B. Inden (2018). An Approach to Sign Language Translation using the Intel RealSense Camera. In 2018 10th Computer Science and Elec-tronic Engineering (CEEC). 219–224.

https://doi.org/10.1109/CEEC.2018.8674227

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

[15] A.D. López (2020). Sign Language Recognition - ASL Recognition with MediaPipe and Recurrent Neural Networks. B.S. thesis. Universitat Politècnica de Catalunya.

[16] M.A. Aowal, A.S. Zaman, S.M.M. Rahman and D. Hatzinakos (2014). Static Hand Gesture Recognition Using Discriminative 2D Zernike Moments. IEEE Region 10 Annual International Conference, Proceedings/TENCON 2015. https://doi.org/10.1109/TENCON.2014.7022345

[17] V. Bheda and D. Radpour (2017). Using Deep Convolutional Networks for Gesture Recognition in American Sign Language. ArXiv abs/1710.06836 (2017).

[18] S.R. Kalbhor and A.M. Deshpande (2018). Digit Recognition Using Machine Learning and Convolutional Neural Network. In 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI). 604–609. https://doi.org/10.1109/ICOEI.2018.8553954

[19] Y.S. Tan, K.M. Lim, C. Tee, C.P. Lee and C.Y. Low (2021). Convolutional neural network with spatial pyramid pooling for hand gesture recognition. Neural Computing and Applications 33 (05 2021), 1–13. https://doi.org/10.1007/s00521-020-05337-0

[20] H. Luqman, M.E. El-Alfy and G. Binmakhashen (2021). Joint space representation and recognition of sign language fingerspelling using Gabor filter and convolutional neural network. Multimedia Tools and Applications 80 (03 2021), 1–22. https://doi.org/10.1007/s11042-020-09994-0

[21] P. Das, T. Ahmed, M.F. Ali (2020). Static Hand Gesture Recognition for American Sign Language using Deep Convolutional Neural Network. 1762– 1765. https://doi.org/10.1109/TENSYMP50017.2020.9230772

[22] M.M. Hasan, A.K. Srizon, A. Sayeed and M.A.M. Hasan (2020). Classification of American Sign Language by Applying a Transfer Learned Deep Convolutional Neural Network.

https://doi.org/10.1109/ICCIT51783.2020.9392703

[23] S. Sharma, H.P.J. Dutta, M. Bhuyan and R. Laskar (2020). Hand Gesture Localization and Classification by Deep Neural Network for Online Text Entry. https://doi.org/10.1109/ASPCON49795.2020.9276713

[24] A.A. Barbhuiya, R.K. Karsh and R. Jain (2021). CNN based feature extraction and classification for sign language. Multimedia Tools and Applications 80 (01 2021), 1–19. https://doi.org/10.1007/s11042-020-09829-y

[25] P.M. The and M.T. Yu (2021). Static and Dynamic Hand Gesture Recognition Using GIST and Linear Discriminant Analysis. Volume 14 (06 2021), 123. https://doi.org/10.22266/ijies2021.0831.12

[26] V. Bazarevsky, Y. Kartynnik, A. Vakunov, K. Raveendran and M. Grundmann (2019). Blazeface: Sub-millisecond neural face detection on mobile gpus. arXiv preprint arXiv:1907.05047 (2019).

[27] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.L. Chang and M. Grundmann (2020). MediaPipe Hands: On-device Real-time Hand Tracking. arXiv:2006.10214 [cs.CV]

[28] T.Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and G. Belongie (2017). Feature pyramid networks for object detection. In Proceed-ings of the IEEE conference on computer vision and pattern recognition. 2117–2125.

[29] T. Simon, H. Joo, I. Matthews and Y. Sheikh (2017). Hand key-point detection in single images using multiview bootstrapping. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 1145–1153.

[30] W.U. Wickramaarachchi and S. Vasanthapriyan, "Multi-Layer Framed Offline Signature Recognition Algorithm," Journal of Image and Graphics, Vol. 3, No. 1, pp. 11-15, June 2015. doi: 10.18178/joig.3.1.11-15