# Bangla Sign Language Detection using SIFT and CNN

Shirin Sultana Shanta
*Computer Science and Engineering*
*American International University-Bangladesh (AIUB)*
Dhaka, Bangladesh
shirin.shanta151094@gmail.com

Saif Taifur Anwar
*Computer Science and Engineering*
*American International University-Bangladesh (AIUB)*
Dhaka, Bangladesh
taifuranowar@gmail.com

Md. Rayhanul Kabir
*Computer Science and Engineering*
*Khulna University*
Khulna, Bangladesh
nabid.ku.cse@gmail.com

*Abstract*—**Very few researches were done in detecting Bangla sign language. Most of the researchers in this field used SVM, ANN or KNN as the classifier. In this paper, we try to implement a Bangla sign language system that uses SIFT feature extraction and Convolutional Neural Network (CNN) for classification. We also show that using SIFT feature increases the accuracy of CNN for detecting Bangla Sign language.**

*Keywords—Bangla Sign Language, SIFT, CNN, BoF.*

## I. INTRODUCTION

In general, deaf people are a very small portion of the general population of Bangladesh. For that, it is very hard for them to do everyday chores of life. They can handle communication via Bangla sign language, but it is not widely known to general people. Technology can enable interpreting Bangla sign language to a person who does not understand it. This is why detecting Bengali sign language is very important for the minor deaf community.

Sign language is one of the manual communications which is very complex in nature. It does not only consist of hand gestures, but also arm movements, head movements, and facial expressions. Bengali sign language is also no exception. A small number of research has been done in the hand gesture recognition of Bengali sign language. Most of them uses SVM as a classifier to detect different alphabets for different hand gestures. But SVM is a generic classifier in machine learning. It is not designed to work with image data in mind. More image specific classifier, like Convolutional Neural Network (CNN) is used in various image processing like optical character or signpost recognition. In this paper, we propose a Bangla sign language detection system which uses SIFT (Scale-Invariant Feature Transform) technique to eliminate any rotational, scaling problems in images and CNN to classify the hand gestures.

## II. RELATED WORK

There were many researches on recognizing hand gesture in Bangla sign language detection but when it comes down to using CNN there were no research at all. One of the pioneering research on Bangla sign language detection was made by Najeefa Nikhat Choudhury and Golam Kayas, who used Artificial Neural Network (ANN) to detect five Bangla hand gesture signs[1]. Rahat Yasir and Riasat Azim Khan proposed a two-handed hand gesture recognition for Bangla sign language[2]. They tested 3 models: model 1 where grayscale images were directly fed to neural network, model 2 where PCA (Principal Component Analysis) was used to classify data, and model 3 where LDA (Linear Discriminant Analysis) was used for feature extraction. Between three models, model 3 yielded best result. Muttaki Hasan, Tanvir Hossain Sajib, and Mrinmoy Dey proposed a hand gesture recognition system by extracting features using HOG (Histogram of Oriented Gradients) and classify gestures using SVM (Support Vector Machine)[3]. Farhad Yasir used SIFT (Scale-Invariant Feature Transform) to extract feature not sensitive to scale, rotation, resolution etc. and then used SVM to classify hand gestures [4]. Muhammad Aminur Rahaman proposed a real time Bangla hand gesture recognition system using KNN (K-Nearest Neighbor) classifier[5].

If we analyze previous work made on Bangla sign language detection, we can see that most popular method to classify hand gesture is SVM, ANN and KNN. No researches were made to classify Bangla sign language hand gestures using CNN (Convolutional Neural Network). But using CNN in recognizing sign language is nothing new. Lionel Pigou used Microsoft Kinect to take depth map and skeleton image of full body and used CNN to classify whole body image to different sign language characters [6]. Jie Huang used 3D convolutional neural network to classify features from raw video stream and then he used those features to classify sign language words[7]. Pavlo Molchanov also used 3D convolutional neural network for real-time sign language detection[8]. They used depth, color and stereo-IR sensors so that the machine can understand rotation of hands and body movements. Tomas Pfister proposed a CNN based system to detect human pose estimation from videos[9]. Their system can work even in different background and it does not harm their accuracy.

## III. METHODOLOGY

Our main classifier for Bangla sign language detection is CNN (Convolutional Neural Network). CNN and SVM shares same image pre-processing and post-processing techniques. But CNN does not need all the preprocessing steps described in our proposed method. But we will show later in the paper that using all the preprocessing techniques before using CNN can improve the performance. But before classifying Bangla sign language, we have used many image pre-processing techniques. We have manually taken 200 images for 38 Bangla signs for 51 Bangla letters. So, in total we have used 7600 images that were used in training and testing.

The whole process can be described into steps:

1. Obtaining the images manually.

2. Implement skin masking technique to crop only Region of interest (ROI) that has only the image of hand.

3. Extract feature descriptors using SIFT (Scale-Invariant Feature Transform)

4. Use k-means clustering to obtain features as clustered descriptors. Use Bag of Features (BoF) to represent the features in histogram of visual vocabulary.

5. Input the data in CNN as histograms and check output for accuracy.

6. Compare the results with SIFT and without SIFT operation.

Figure 1 shows the flowchart of the proposed process. We have not used any automated image collection system where the training and testing data can be obtained from webcam. The image database was created manually.

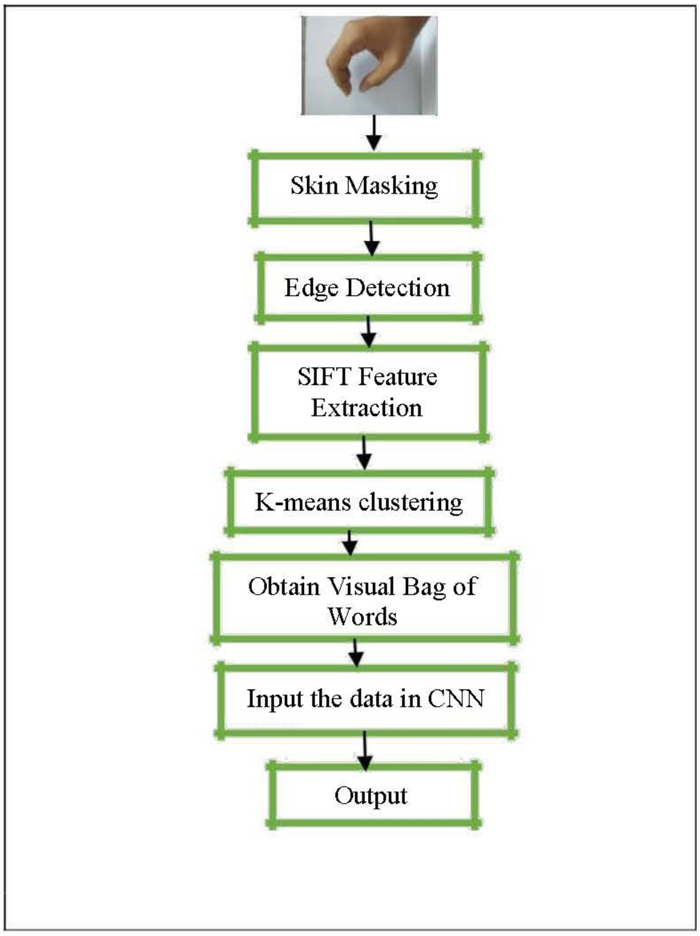


Fig. 1. Flowchart of the proposed method

We also used one handed gestures but in Bangla sign language two handed gestures are also available. We used openCV and python for image pre-processing and classification techniques.

## IV. PRE-PROCESSING

### A. Skin Masking

The training images needs to be pre-processed before they can be used in CNN or SVM. CNN or SVM does not like noise or extra data. So, we have used skin masking technique to crop only Region of interest (ROI) that has only the image of hand.

The training images were in RGB color space. We have used openCV function to convert from RGB to HSV. After the images are in HSV space, we need to set lower boundary and upper boundary to do skin masking. This upper and lower boundary make sure that all other colors except skin color is deleted from the image. Figure 2 shows the output in openCV after applying upper and lower bounds.

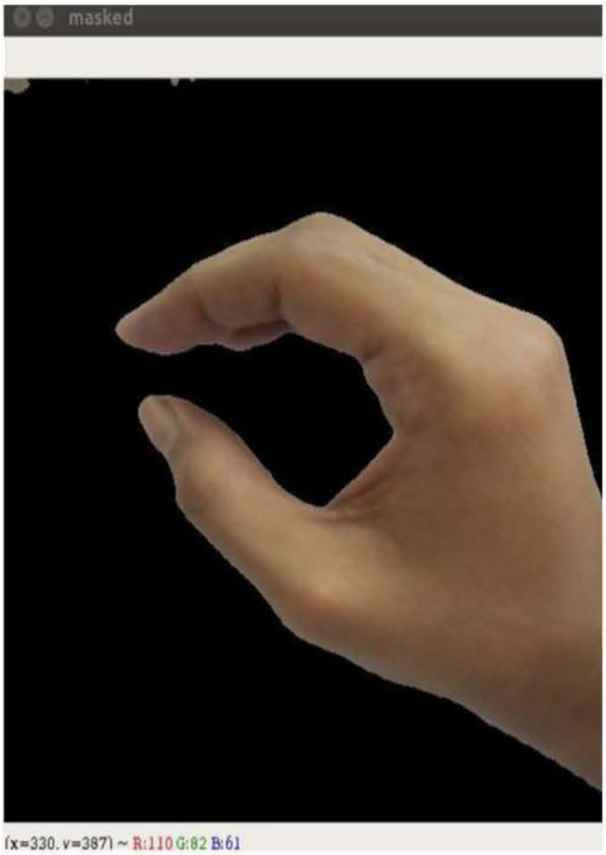
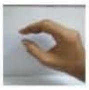
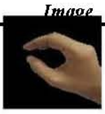
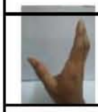
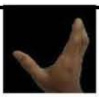
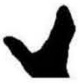
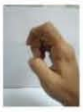
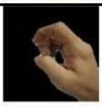
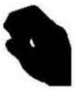
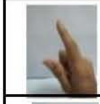
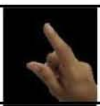
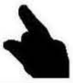
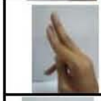
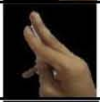
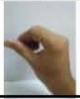
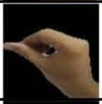
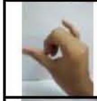
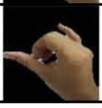
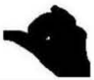
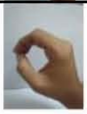
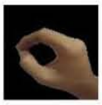
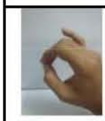
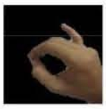
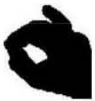
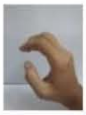
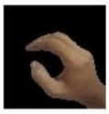
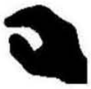
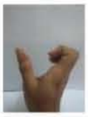
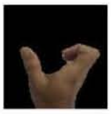
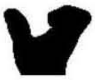
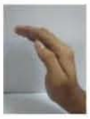
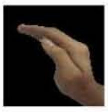
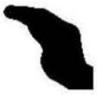
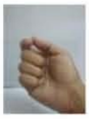
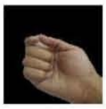
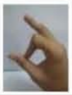
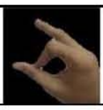
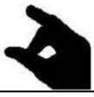
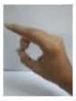
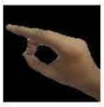
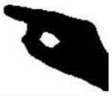
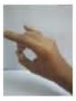
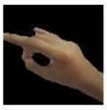
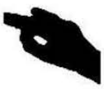
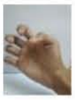
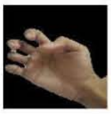
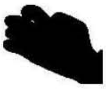
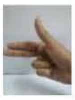
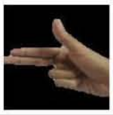
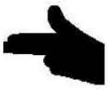
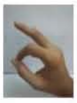
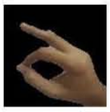
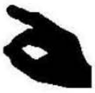
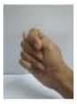
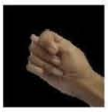
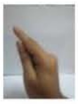
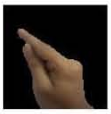
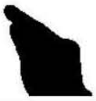
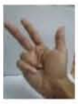
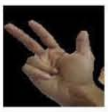
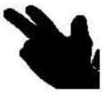
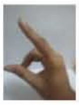
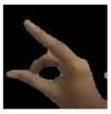
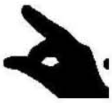
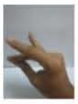
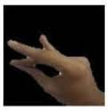
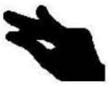
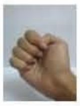
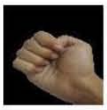
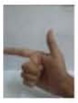
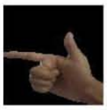
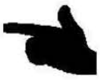
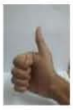
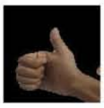
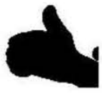


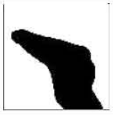
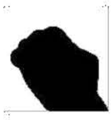
Fig. 2. Applying skin mask in Bangla sign language letter অ / য়

After the conversion, pixels that are white indicates skin area and pixels that are black indicates not skin area. However, this yields some false positive which should be removed. We used elliptical kernel to perform erosion and dilations. These actions make sure that false positives are removed from output images.

After skin area is left on the image, we converted it to grayscale before moving into next step. Table 1 shows all Bangla signs and its corresponding letter(s) as the output, where false positives are nonexistent and the images are in grayscale.

TABLE I. SKIN MASKED AND GRAYSCALED IMAGES

| **Original image** | **Converted Images** | | |
|---|---|---|---|
| | ***Skin Masked Image*** | ***Grayscale Image*** | ***Bangla letter*** |
|  |  | | অ/য় |
|  |  |  | আ / ◌া |
|  |  |  | ই / ঈ / ি◌া◌া / ◌া |
|  |  |  | উ/ঊ/ ◌া / ◌া |
|  |  | | ঋ/ র /ড় /ঢ় / ◌া |
|  |  | | এ/ ে◌◌া◌া |
|  |  |  | ঐ/ ৈ◌◌া◌া |
|  |  | | ও / ে◌◌া◌া |
|  |  |  | ঔ/ ে◌◌া◌া |
|  |  |  | ক |
|  |  |  | খ/ ক্ষ |
|  |  |  | গ |
|  |  | | ঘ |
|  |  |  | ঙ |
|  |  |  | চ |
|  |  |  | ছ |
|  |  |  | জ/ য |
|  |  |  | ঝ |
|  |  |  | ঞ |
|  |  | | ট |
|  |  |  | ঠ |
|  |  |  | ড |
|  |  |  | ঢ |
|  |  |  | ন/ণ |
|  |  | | ত/ৎ |
|  |  |  | থ |
|  |  |  | দ |

| Original image | Converted Images | | |
|---|---|---|---|
| | ***Skin Masked Image*** | ***Grayscale Image*** | ***Bangla letter*** |
| | | | ধ |
| | | | প |
| | | | ফ |
| | | | ব/ভ |
| | |  | ম |
| | | | ল |
| | |  | শ/স/ষ |
| | | | হ |
| | | | ◌া |
| | | | ◌া |
| | | | ঁ |

### B. *Canny Edge Detection*

After Region of interest was found, it was time to detect the edge of the hand in the image. The most popular edge detection algorithm called Canny Edge Detection, according to openCV documentation [10], was used to detect the edges of the hand. Canny Edge technique looks for sharp discontinuities in an image, which is a major feature of edges.

### C. *Extracting feature by SIFT (Scale-Invariant Feature Transform)*

One of the major problems in training images is many classification methods are susceptible to rotation, scaling, resolution, changes in illumination etc. So we did not use images directly to CNN. Rather, we used a technique called SIFT (Scale-Invariant Feature Transform). We have implemented SIFT with following steps:

**1. Keypoint detection:** First we search for rapid intensity changes in the image. The search is done in both horizontal and vertical direction (x,y). Such pixels are referred to as octave, and when we find one we save its keypoints as (x,y) coordinates. Another variable σ is used which represents the Gaussian scale space where the keypoints exist. So, we can find the rapid intensity changes across the scale and space which gives us a matrix of (x,y,σ) values.

**2. Keypoint descriptor generation:** After getting keypoints, we computed gradient of each pixel. The gradient can be described as intensity changes as x and y direction changes. Then SIFT bins the gradients to 8 directions (N, NE, E, SE, S, SW, W, NW). Thus, we can make 8-bit histogram. We divide the image into 16 regions, each region has a histogram. We then concatenate to get 16 regionals 8-bit histogram, or in another term, 128 elements SIFT descriptor matrix. In Figure 3 we can see the visual output the SIFT function and where it has found the features.

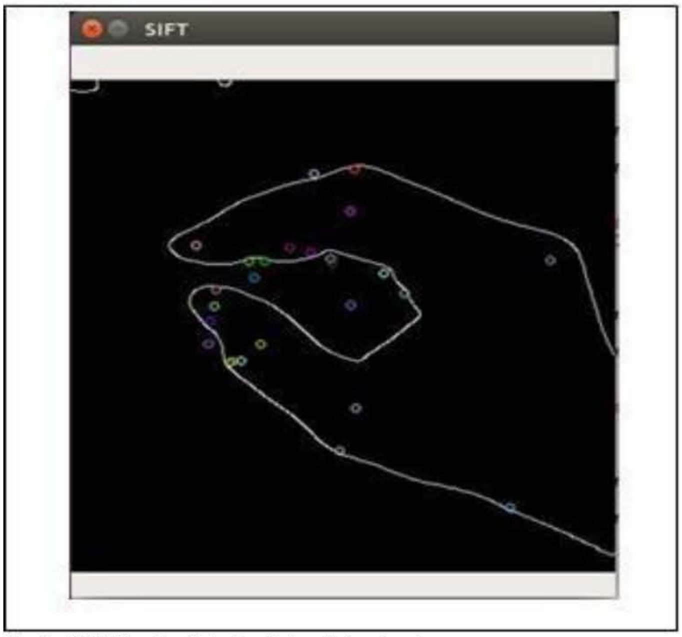


Fig. 3. SIFT function detecting feature in hand gesture

SIFT descriptors has random number of descriptors with 128 dimensions. CNN needs a fixed amount of feature vectors to work. So, we clustered the features using K-means clustering. A total of 150 clusters was found using k-means clustering, which indicates 150 features of our image dataset. We then converted each image as a histogram, where in x axis we have put 150 features and in y axis we have put the frequency of each features in a given image. That way we represented each image in histogram by frequency of occurrence of all clustered features. This is the basis of our Bag of Feature set. We wanted to classify 38 Bangla signs. BoF represents each image as a histogram of features. In our image framework the histogram of 38 classes of sign language was generated. There are 200 images each for 38 Bangla signs, so in total 7600 histograms were generated to classify those Bangla signs. Please note that the frequency for each 150 features in the histogram is random. So, we checked each histogram size and found max frequency as 200. So, in next step, histograms with matrix size of 150 x 200 was used.

## V. IMPLEMENTING CNN

Convolutional neural network (CNN) is a feed-forward artificial neural network that is successfully applied to the field of image processing. It has convolutional layers, pooling layers, dropout layers and normalization layers. We have chosen CNN as it usually gives better result even without preprocessing. Our SIFT based image preprocessing should give it an extra boost.

In figure 4 we have shown the architecture of our CNN. It has 4 convolutional layer and as activation function, ReLU is used. As we have 150 features and maximum 200 frequency in histogram, matrix of 150 x 200 was used as input. Our CNN finalizes the output by having 2 fully connected layers. First layer has 128 neurons and second layer has 38 neurons. The 38 neurons correspond to 38 classes or Bangla sign language signs. The 128 neurons are SIFT descriptors. Some dropout layers are used to prevent overfitting[11].

We have used stochastic gradient descent optimizer [12] and set the learning rate to 0.5 and momentum as 0.1. If we look at the model at Figure 5 we can see that convolutional layer size decreases as it is closer to the output. This is to maximize the accuracy of the CNN[13].

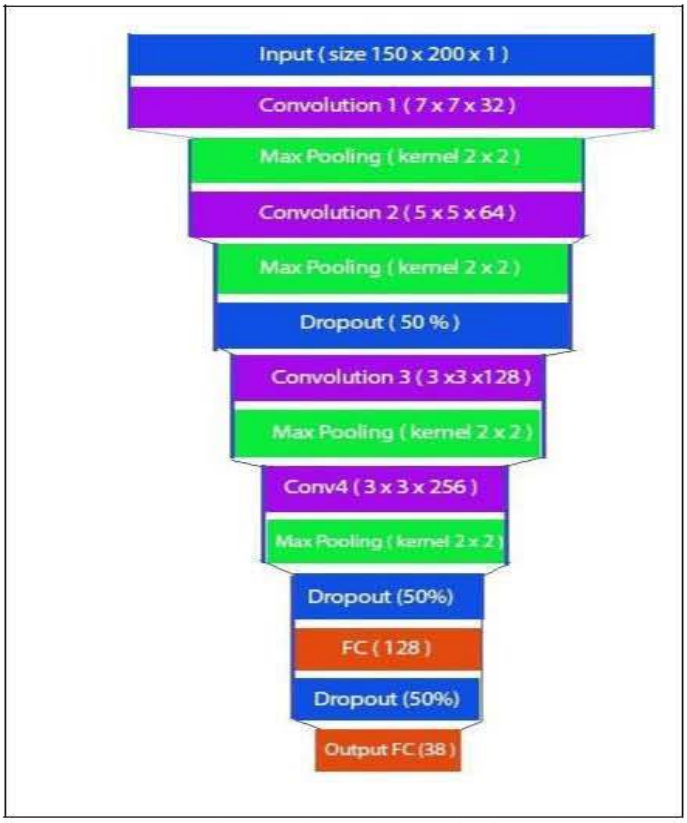

Input ( size 150 x 200 x 1 )
Convolution 1 ( 7 x 7 x 32 )
Max Pooling ( kernel 2 x 2 )
Convolution 2 ( 5 x 5 x 64 )
Max Pooling ( kernel 2 x 2 )
Dropout ( 50 % )
Convolution 3 ( 3 x3 x128 )
Max Pooling ( kernel 2 x 2 )
Conv4 ( 3 x 3 x 256 )
Max Pooling ( kernel 2 x 2 )
Dropout (50%)
FC ( 128 )
Dropout (50%)
Output FC (38)

Fig. 4. Our proposed CNN architecture for Bangla Sign Language detection

## VI. EXPERIMENTAL RESULTS

This paper included 38 Bangla signs, which can be used in 51 Bangla letters. This sign languages was gathered from Bangla Sign Language Dictionary[14]. We did not include two-handed gesture as this is the first time CNN was applied to Bangla sign language. The number of training and testing samples were 1700 in total. Although it seems very high, for good accuracy and real-life testing, much larger dataset is needed. But this is a small step in the direction of implementing CNN. In table 2 we can see the results with SIFT disabled and SIFT enabled results. We can see when SIFT is enabled, there is a vast amount of accuracy increase. This is due to the fact that SIFT helps CNN to extract features without scaling, rotation, resolution problem. Although CNN proves better against those problems, these results proves that using SIFT with CNN yields better results.

TABLE II. CNN CLASSIFICATION RESULT

| **Bangla Vowel & consonants** | **CNN accuracy** | |
|---|---|---|
| | ***With SIFT*** | *Without SIFT* |
| অ/য় | 90% | 78% |
| আ / ◌া | 93% | 72% |
| ই / ঈ / ি◌া / ◌া | 88% | 70% |
| উ/ঊ/ ◌া / ◌া | 89% | 77% |
| ঋ/ র /ড় /ঢ় / ◌া | 91% | 75% |
| এ/ ে◌◌া | 92% | 73% |
| ঐ/ ৈ◌◌া | 90% | 78% |
| ও / ে◌◌া | 87% | 75% |
| ঔ/ ে◌◌া | 88% | 80% |
| ক | 89% | 82% |
| খ/ক্ষ | 90% | 79% |
| গ | 92% | 77% |
| ঘ | 90% | 81% |
| ঙ | 93% | 88% |
| চ | 88% | 81% |
| ছ | 91% | 78% |
| জ/ য | 90% | 77% |
| ঝ | 93% | 79% |
| ঞ | 94% | 81% |
| ট | 88% | 78% |
| ঠ | 90% | 77% |
| ড | 92% | 81% |
| ঢ | 91% | 80% |
| ন/ণ | 93% | 82% |
| ত/ৎ | 90% | 78% |
| থ | 93% | 77% |

| Bangla Vowel & consonants | CNN accuracy | |
|---|---|---|
| | *With SIFT* | *Without SIFT* |
| দ | 88% | 71% |
| ধ | 87% | 73% |
| প | 90% | 77% |
| ফ | 92% | 88% |
| ব/ভ | 92% | 77% |
| য | 90% | 78% |
| ল | 93% | 75% |
| শ/স/ষ | 95% | 88% |
| হ | 90% | 75% |
| ◌া | 89% | 77% |
| ◌া | 90% | 77% |
| ৎ | 93% | 85% |

## CONCLUSION

In this paper, we have shown 38 Bangla sign detection using SIFT and CNN. In the Bangla sign language detection research, various classification techniques like SVM / ANN was used but CNN was never tested. We have shown that with SIFT inputs, CNN works well in Bangla sign language detection. We have taken all the training and testing images in fixed illumination. So we also assume the system which will use our framework would control input image of our framework and handles illumination properly. Our input images, which consists image of hands, are not symmetric shape by any nature. So our framework is safe from SIFT's illumination problem and symmetric shape detection failure. However, there are two-handed gestures in Bangla sign language that it cannot detect. Also, there would be speed improvement if SURF was used instead of SIFT. On the other hand, this system currently isn't able to detect full body gestures. There are many improvements to be made to the Bangla sign language detection. However, the proposed system yielded a better result than expected and this fueled us to continue more investigation on improving the CNN accuracy. CNN is already popular in various image processing systems and we believe it can also make a huge impact on Bangla sign language. We hope to continue improving upon the proposed system.

## ACKNOWLEDGMENT

The authors would like to thank the department of Computer Science and Engineering of American International University-Bangladesh (AIUB).

## REFERENCES

[1] Choudhury, Najeefa Nikhat, and Golam Kayas. *Automatic recognition of Bangla sign language*. Diss. BRAC University, 2012.

[2] Yasir, Rahat, and Riasat Azim Khan. "Two-handed hand gesture recognition for Bangla sign language using LDA and ANN." *Software, Knowledge, Information Management and Applications (SKIMA), 2014 8th International Conference on*. IEEE, 2014. I. S. Jacobs and C. P. Bean,
"Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[3] Hasan, Muttaki, Tanvir Hossain Sajib, and Mrinmoy Dey. "A machine learning based approach for the detection and recognition of Bangla sign language." *Medical Engineering, Health Informatics and Technology (MediTec), 2016 International Conference on*. IEEE, 2016.

[4] Yasir, Farhad, et al. "Sift based approach on bangla sign language recognition." Computational Intelligence and Applications (IWCIA), 2015 IEEE 8th International Workshop on. IEEE, 2015.

[5] Rahaman, Muhammad Aminur, et al. "Real-time computer vision-based Bengali Sign Language recognition." Computer and Information Technology (ICCIT), 2014 17th International Conference on. IEEE, 2014.

[6] Pigou, Lionel, et al. "Sign language recognition using convolutional neural networks." Workshop at the European Conference on Computer Vision. Springer, Cham, 2014.

[7] Huang, Jie, et al. "Sign language recognition using 3d convolutional neural networks." Multimedia and Expo (ICME), 2015 IEEE International Conference on. IEEE, 2015.

[8] Molchanov, Pavlo, et al. "Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[9] Pfister, Tomas, et al. "Deep convolutional neural networks for efficient pose estimation in gesture videos." *Asian Conference on Computer Vision*. Springer, Cham, 2014.

[10] Opencv-python-tutroals.readthedocs.io. (2018). *Canny Edge Detection — OpenCV-Python Tutorials 1 documentation*. [online] Available at: http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_canny/py_canny.html [Accessed 31 Mar. 2018].

[11] Srivastava, Nitish, et al. "Dropout: A simple way to prevent neural networks from overfitting." *The Journal of Machine Learning Research* 15.1 (2014): 1929-1958.

[12] Shamir, Ohad, and Tong Zhang. "Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes." *International Conference on Machine Learning*. 2013.

[13] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

[14] Bangla Sign Language Dictionary, 2nd ed., Bangladesh Sign Language Committee, 1997, pp. 92-117.