Bangla Sign Language Detection Using CNN & OpenCV Image Processing

A project report

Submitted by

Mohammed Asfaqul Alam
Reg: 2018338063

Moshaffek Ahmed Sidaf
Reg: 2018338065

Montasin Ahmed Uzan
Reg: 2018338066

Sahadat Hossain
Reg: 2018338067

Md. Yusuf
Reg: 2018338068

Submitted To

Md. Ishfak Tahmid

Assistant Professor

In partial fulfillment for the course_Project of EEE 332

Electrical & Electronics Engineering

Shah Jalal University of Science & Technology

Sylhet, Bangladesh

Table of Contents

Bangla Sign Language Detection Using

CNN & OpenCV Image Processing

Abstract

For hearing and speaking impaired community sign language is the best way for communication. It is usually hard for most people who are not familiar with sign language to communicate without an interpreter. In Bangladesh about 2.7 million people who are unable to interact with people using sign language. So, for this reason we try to build a system for Bangladeshi citizens with the help of Deep learning & Artificial intelligence to make the sign language to text language. The technique that has been implemented here, transcribes the gesture from sign language to spoken language which is easily understood by the any listening. Our framework employs convolutional neural networks (CNN) to learn from the images in our dataset and interpret 27 Bangla Alphabet signs from input images.

Our system takes snapshots from a video by using a webcam with applying a computer vision-based approach. After that, it compares those input photos to a previously trained dataset generated with CNN and displays the Bengali alphabets. After estimating the model on our dataset, we obtained an overall accuracy of 92%.

Keywords & Abbreviation

Bangla Sign language (BSL),

Convolutional Neural Network (CNN),

Training, Dataset, Gaussian Filters, Machine Learning,

OpenCV, Image Processing, Model accuracy,

Python 3.0

Introduction

Sign language is a natural language used by hearing and speech impaired people to communicate. A sign language interpreter is a significant step toward improving contact between the deaf and the general population. There arise needs for sign translators which can translate sign language to spoken Bangla language. Our model uses hand gestures instead of sound to convey messages or information.

In recent development in the area of deep learning, Neural networks may have far reaching implications and implementations for sign language analysis. In our proposed system, Convolutional Neural Networks (CNN) is used to classify images of sign language as CNN are faster in feature classification of images over other classifiers. It sensed the symbol with only one hand. It split into two parts: the trained and the signs detection part.

We make use of Convolutional Neural Networks (CNN) and various image processing technique and computer vision for training and to classify the images. We are able to recognize 27 Bangla sign gesture alphabets with high accuracy. Our model has achieved a remark able accuracy of above 80%

We mainly follow these three steps

- Creating the dataset using Image processing technique
- Training CNN on the captured dataset
- Predicting the dataset

Literature Review

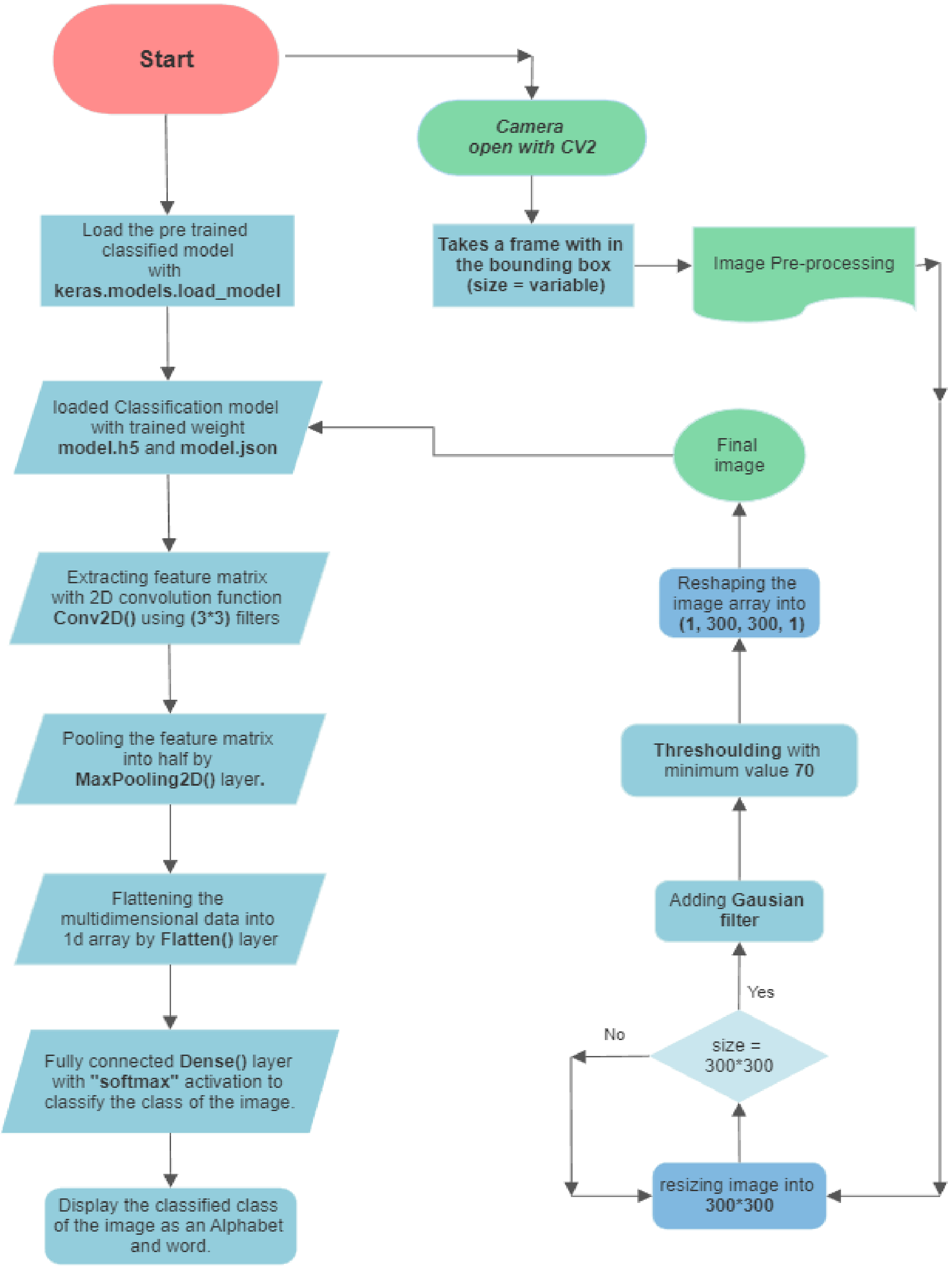
This research suggested the use of filters in sign language translation algorithm because the existing system has low accuracy as it faced issue with skin tone identification. Sign language conversion can reach a maximum of 96% of accuracy but achieving that can be tedious task. The current system has some drawbacks to identify the skin tone under low light areas.

Objective

The key goal is to recognize the sign with maximum accuracy apart from different light, dark conditions must be developed.

Methodology

This proposed system takes the images from a camera and then preprocesses the images. To identify the image structure firstly the system applies transformation on images, then it applies extra light effects on images. After picture is taken, it measures the picture dimension. As dimension is done then it transforms it into a binary image. It uses deep learning techniques to construct a qualified dataset. Classifies with the qualified dataset after the recognizer section and identifies the alphabet of the Bangla language. The entire proposed model displayed by a flow chart below;



Start
Camera open with CV2
Load the pre trained classified model with keras.models.load_model
Takes a frame with in the bounding box (size = variable)
Image Pre-processing
loaded Classification model with trained weight model.h5 and model.json
Final image
Extracting feature matrix with 2D convolution function Conv2D() using (3*3) filters
Reshaping the image array into (1, 300, 300, 1)
Pooling the feature matrix into half by MaxPooling2D() layer.
Thresholding with minimum value 70
Flattening the multidimensional data into 1d array by Flatten() layer
Adding Gausian filter
Yes
No
size = 300*300
Fully connected Dense() layer with "softmax" activation to classify the class of the image.
Display the classified class of the image as an Alphabet and word.
resizing image into 300*300

Data Collection

We collect images from the video frame using the webcam of our laptop. We use Intel i5 @3.6GHz laptop with 4 GB ram and NVidia MX130 GPU for running the system.

Our dataset consists of 27 hand signs and a total of more than 50000 raw pictures which consists various lengths, orientation and intensities.

Image Processing

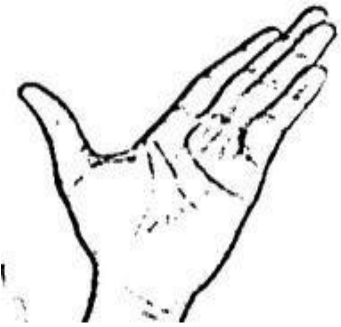
The software we use of this system is python 3.0, OpenCV, Tensorflow and Numpy. Among all programming language python is the fastest, it used by this system.

Tensorflow is an open source machine learning tool that is used to train the sign images from start to finish.

We also use Gaussian filter in our proposed model. Gaussian filter is good for smoothen the sharpness of the edges in an image and remove noise.

Thresholding is one of the most common segmentation techniques in computer vision and it allows us to separate the foreground from the background of the image. We use **cv2.threshold** function to apply basic thresholding.

Raw image



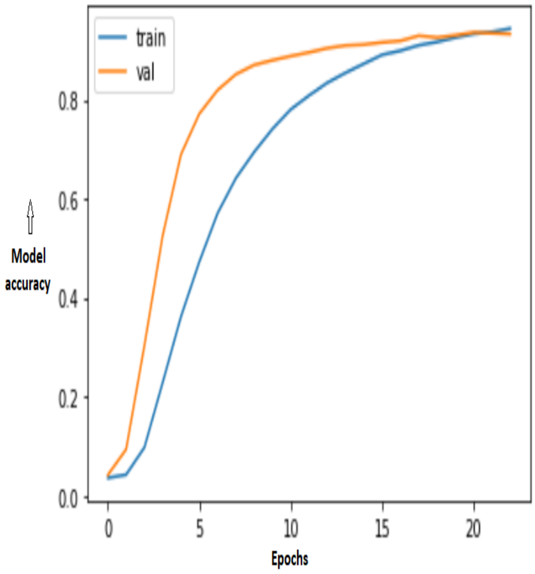
Processed Image

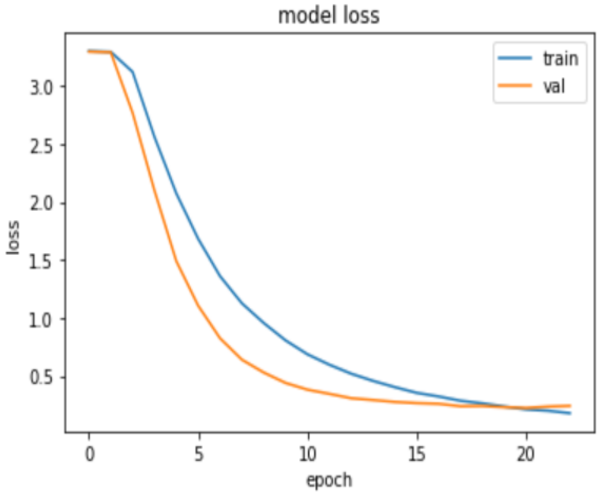
Result & Discussion

To develop the area of sign language interpretation, our proposed model has variety of approaches. We developed the proposed model to improve the technique on Bengali alphabetical sign detection. The accuracy of each sign's identification is calculated by this method for experimental results. Formula's we used are shown below:

Precision = $\frac{Tp}{Tp+Fp}$

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



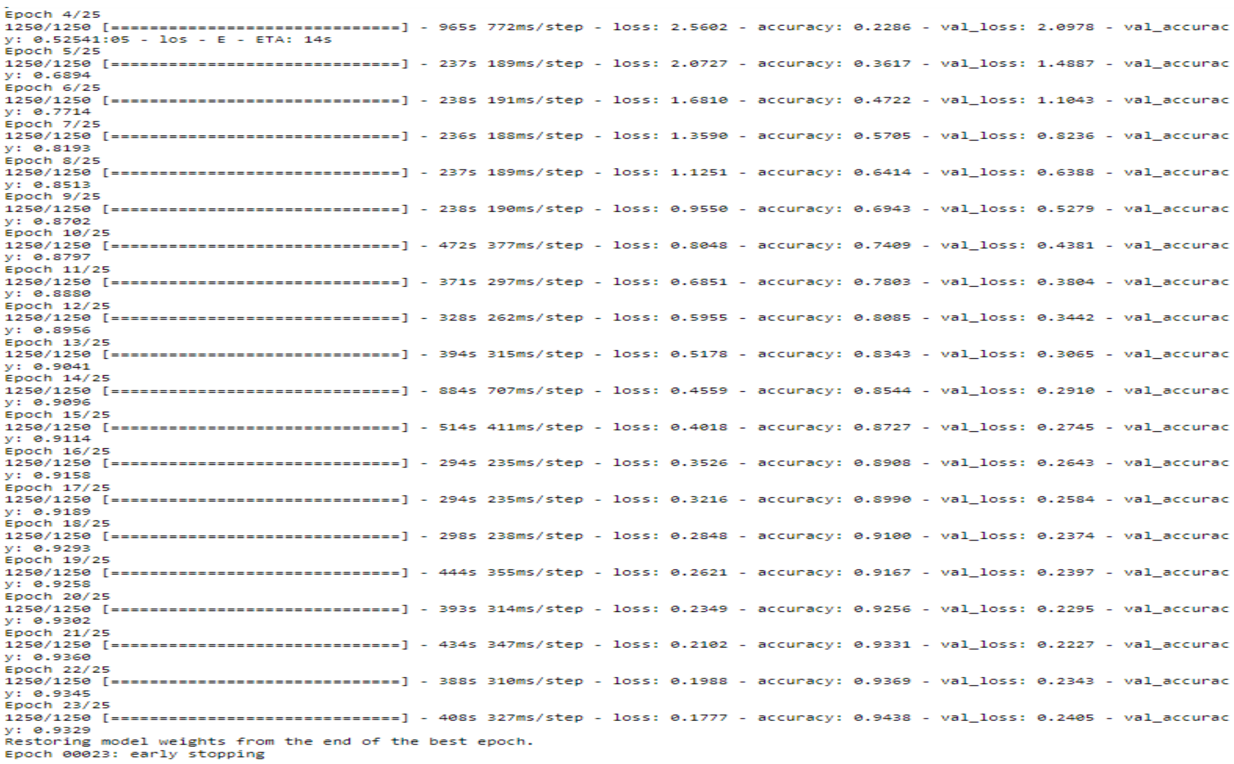


```
<Figure size 432x288 with 0 Axes>
```

Fig 1

Fig 2

Figure 1 shows how the system uses 27 tests on hand signs for each alphabet and correctly recognize from these images. Our model for the Bengali sign language detection achieved 94% for training, 95% validation, 93.29% for training.



```
Epoch 4/25
1250/1250 [==============================] - 965s 772ms/step - loss: 2.5602 - accuracy: 0.2286 - val_loss: 2.0978 - val_accurac
y: 0.52541:05 - los - E - ETA: 14s
Epoch 5/25
1250/1250 [==============================] - 237s 189ms/step - loss: 2.0727 - accuracy: 0.3617 - val_loss: 1.4887 - val_accurac
y: 0.6894
Epoch 6/25
1250/1250 [==============================] - 238s 191ms/step - loss: 1.6810 - accuracy: 0.4722 - val_loss: 1.1043 - val_accurac
y: 0.7714
Epoch 7/25
1250/1250 [==============================] - 236s 188ms/step - loss: 1.3590 - accuracy: 0.5705 - val_loss: 0.8236 - val_accurac
y: 0.8193
Epoch 8/25
1250/1250 [==============================] - 237s 189ms/step - loss: 1.1251 - accuracy: 0.6414 - val_loss: 0.6388 - val_accurac
y: 0.8513
Epoch 9/25
1250/1250 [==============================] - 238s 190ms/step - loss: 0.9550 - accuracy: 0.6943 - val_loss: 0.5279 - val_accurac
y: 0.8702
Epoch 10/25
1250/1250 [==============================] - 472s 377ms/step - loss: 0.8048 - accuracy: 0.7409 - val_loss: 0.4381 - val_accurac
y: 0.8797
Epoch 11/25
1250/1250 [==============================] - 371s 297ms/step - loss: 0.6851 - accuracy: 0.7803 - val_loss: 0.3804 - val_accurac
y: 0.8880
Epoch 12/25
1250/1250 [==============================] - 328s 262ms/step - loss: 0.5955 - accuracy: 0.8085 - val_loss: 0.3442 - val_accurac
y: 0.8956
Epoch 13/25
1250/1250 [==============================] - 394s 315ms/step - loss: 0.5178 - accuracy: 0.8343 - val_loss: 0.3065 - val_accurac
y: 0.9041
Epoch 14/25
1250/1250 [==============================] - 884s 707ms/step - loss: 0.4559 - accuracy: 0.8544 - val_loss: 0.2910 - val_accurac
y: 0.9096
Epoch 15/25
1250/1250 [==============================] - 514s 411ms/step - loss: 0.4018 - accuracy: 0.8727 - val_loss: 0.2745 - val_accurac
y: 0.9114
Epoch 16/25
1250/1250 [==============================] - 294s 235ms/step - loss: 0.3526 - accuracy: 0.8908 - val_loss: 0.2643 - val_accurac
y: 0.9158
Epoch 17/25
1250/1250 [==============================] - 294s 235ms/step - loss: 0.3216 - accuracy: 0.8990 - val_loss: 0.2584 - val_accurac
y: 0.9189
Epoch 18/25
1250/1250 [==============================] - 298s 238ms/step - loss: 0.2848 - accuracy: 0.9100 - val_loss: 0.2374 - val_accurac
y: 0.9293
Epoch 19/25
1250/1250 [==============================] - 444s 355ms/step - loss: 0.2621 - accuracy: 0.9167 - val_loss: 0.2397 - val_accurac
y: 0.9258
Epoch 20/25
1250/1250 [==============================] - 393s 314ms/step - loss: 0.2349 - accuracy: 0.9256 - val_loss: 0.2295 - val_accurac
y: 0.9302
Epoch 21/25
1250/1250 [==============================] - 434s 347ms/step - loss: 0.2102 - accuracy: 0.9331 - val_loss: 0.2227 - val_accurac
y: 0.9360
Epoch 22/25
1250/1250 [==============================] - 388s 310ms/step - loss: 0.1988 - accuracy: 0.9369 - val_loss: 0.2343 - val_accurac
y: 0.9345
Epoch 23/25
1250/1250 [==============================] - 408s 327ms/step - loss: 0.1777 - accuracy: 0.9438 - val_loss: 0.2405 - val_accurac
y: 0.9329
Restoring model weights from the end of the best epoch.
Epoch 00023: early stopping
```

Figure 3

The number of epochs is a hyperparameter that defines the number times that learning algorithm will work through the entire dataset. Fig 3 shows the number of epochs we takes for our model.

The number of epochs is traditionally large, often hundreds or thousands, allowing the learning algorithm to run until the error from the model has been minimized. As from fig 2, the error in model decreased when the number of epochs increasing and losses in the model is significantly low.

From above result, it is shown that the machine is capable of understanding the plurality of signals. On all of the signs, the highest precision is 90%.

Conclusion

Our proposed system successfully predicts the signs of sign. It uses CNN for training and classification of images. For classification and training, more informative features from the images are finely extracted and being used. A total of 50,000 static images for each sign are used for training to get the accurate result or output. Finally, the output of recognized sign is shown in from of Text as well as converted into Bangla speech.

Thus, this is a user friendly system that can be easily accessed by all the hearing and speech impaired people.

Reference

[1]. S. C. W. Ong and S. Ranganath, — "Automatic sign language analysis: A survey and the future beyond lexical meaning," IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 6, pp. 873– 891.

[2]. M. Hasan, T. H. Sajib and M. Dey, "A machine learning based approach for the detection and recognition of Bangla sign language," 2016 International Conference on Medical Engineering, Health Informatics and Technology (MediTec), 2016, pp. 1-5, doi: 10.1109/MEDITEC.2016.7835387.

[3]. F.M. Javed Mehedi Shamrat, Asaduzzaman, A. K. M. Sazzadur Rahman, R. T. H. Tusher, and Zarrin Tasnim, “A Comparative Analysis of Parkinson Disease Prediction Using Machine Learning Approaches,” International Journal of Scientific & Technology Research, vol. 8, no. 11, pp: 2576-2580, 2019.

[4]. F. M. Javed Mehedi Shamrat1, Sovon Chakraborty2, Md. Masum Billah3, Moumita Kabir4, Nazmus Shakib Shadin5, Silvia Sanjana, "Bangla numerical sign language recognition using convolutional neural networks," Indonesian Journal of Electrical Engineering and Computer Science Vol. 23, No. 1, pp. 405~413

[5]. M. M. Hasan, A. Y. Srizon, A. Sayeed and M. A. M. Hasan, "Classification of Sign Language Characters by Applying a Deep Convolutional Neural Network," 2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT), 2020, pp. 434-438, doi: 10.1109/ICAICT51780.2020.9333456.

[6]. M. A. Rahaman, M. Jasim, M. H. Ali and M. Hasanuzzaman, "Real-time computer vision-based Bengali Sign Language recognition," 2014 17th International Conference on Computer and Information Technology (ICCIT), 2014, pp. 192-197, doi: 10.1109/ICCITechn.2014.7073150.