An Efficient Sign Language Translator Device Using Convolutional Neural Network and Customized ROI Segmentation

Conference Paper · April 2019

DOI: 10.1109/ICCET.2019.8726895

4 authors, including:

Saleh Khan
American International University-Bangladesh
2 PUBLICATIONS **4** CITATIONS

SEE PROFILE

An Efficient Sign Language Translator Device Using Convolutional Neural Network and Customized ROI Segmentation

Saleh Ahmad Khan
Department of Electrical and Electronics Engineering
American International University-Bangladesh
Dhaka, Bangladesh
e-mail: salehahmad.aiub@gmail.com

S. M. Asaduzzaman
Department of Electrical and Electronics Engineering
American International University-Bangladesh
Dhaka, Bangladesh
e-mail: mugdho95@gmail.com

Amit Debnath Joy
Department of Electrical and Electronics Engineering
American International University-Bangladesh
Dhaka, Bangladesh
e-mail: amitdebnathjoy@gmail.com

Morsalin Hossain
Department of Electrical and Electronics Engineering
American International University-Bangladesh
Dhaka, Bangladesh
e-mail: morsalinmunna14@gmail.com

***Abstract*—Sign language is widely used by hearing impaired people all over the world. With the advancement of cutting-edge deep learning techniques, there has been immense attention given by the researchers for sign language conversion. But only a few works have been executed on the Bangla Sign Language conversion for hearing impaired people. This paper aims to demonstrate a user-friendly approach towards Bangla Sign language to text conversion through customized Region of Interest (ROI) segmentation and Convolutional Neural Network (CNN). 5 sign gestures are trained using custom image dataset and implemented in Raspberry Pi for portability. Using the ROI selection approach, the process shows better outcomes than conventional approaches in terms of accuracy level and real time detection from video streaming through webcam. Furthermore, this method serves to offer an efficient model which ultimately results in easy addition of more signs to the final prototype made using Raspberry Pi.**

Keywords-Bangla sign recognition; convolutional neural network; data augmentation; data preprocessing; raspberry Pi

I. INTRODUCTION

In recent years, many works have been executed on image classification task incorporating embedded systems and machine learning algorithms. More often, researchers preferred to use deep learning techniques in their work which eventually outperformed conventional feature extraction-based algorithms and showed better precision and accuracy. A gesture-based system [1] was proposed where flex sensor mounted glove was made to show individual gestures and prerecorded audio was played when certain gesture is made. For processing the sensory data and audio, Arduino Nano was used by researchers of this work. However, these types of glove with wires and sensors connected to it, is not user friendly for mute and dumb people.

With the rapid development of vision-based algorithms researchers come up with image processing-based system to make sign language recognition more user compatible and feasible. One such work [2] presents a methodology of unique feature extraction and matching those with available dataset of templates. It identifies sign up to a threshold value which implies the maximum difference value between the given sign and database. However, in various brightness and backgrounds this method does not stand up to the mark.

Therefore, recently deep learning-based approaches have been adopted for sign language recognition which performed surprisingly well in object detection other computer vision problems. In many cases, they followed CNN model by varying its layer parameters to detect and label signs with the help of a low-cost camera [3-5]. But in case of scalability to practical cases, they did not work well as they required huge number of images for each sign as training data. the training also required a lot of time to learn the image features from the edges to richer set of features. In another work, Natural Language Processing (NLP) based techniques are followed which actually uses HaarCascade classifier for sign identifying and CamShift algorithm for tracking the sign [6]. Words for respective signs are sent to POS tagger module and by using LALR parser meaningful sentence for certain sign is generated. This method serves well result with 90% accuracy claimed by the researchers but it also consists of the drawback of poor detection as HaarCascade failed to detect object in low luminosity condition. In addition to that, a total of 7 steps are required to generate sentence from the video of sign which comprised of a lot of computationally heavy workflow altogether.

To alienate these aforesaid issues, a reverse engineering method is adopted in which a bounding box will be present on display before the classification starts and user need to move that bounding box to that area where sign is made by hearing impaired individual. Only the region inside the bounding box is sent to the trained CNN model for prediction. The main advantage of this process is CNN does not need to learn a lot of features and detect the ROI. With

only a small amount of data it provides much accuracy and faster detection rate. Along with that, with hardware integration using Raspberry Pi it provides much flexibility and scalability for deaf and mute people who uses Bangla Sign Language.

II. Proposed Methodology

The method adopted in this work is segmented into two parts. At first CNN is trained with training set images and later the trained model is implemented in webcam connected Raspberry Pi which finally perform the task of sign detection and labeling on the display connected with the Pi.

A. Bangla Sign Language Learning

At the very beginning, custom image dataset is created as training set for CNN. Data are preprocessed and augmented and finally resized into 96×96 size. By using scikit label encoder the labels for each class are normalized. The encoded files are saved as .npz file for faster processing of training set into the network. The convolutional neural network configuration consists of three convolutional layer two fully connected layer. To normalize the features of layers, batch normalization technique is used with each convolutional layer. After the completion of training, the model is tested with new test set of images which is unknown to the network to come up with unbiased accuracy result.

B. Hardware Integration

Proper hardware association with the trained and tested model was necessary to ensure the portability and stability of the device in terms of the users' perspective. Comparatively low-cost Raspberry pi 3B is considered for this case and it resulted in quite good combination while maintain the same time and precision level previously tested on a computer.

III. Experimental Overview of Design

In this section, key portions of the design which includes modeling of the whole system and the CNN architecture used for the proposed system is explicitly described. Since the main concern of this work was to reduce the training time as well as increase accuracy rather than conventional approaches so data preprocessing was very important for reducing overfitting and under-fitting.

A. Modeling of Sign to Text Conversion

The initial stage was creating the image dataset for 5 signs of Bangla sign language. As the data obtained for each sign were not significant in number and diverse enough to fed into the CNN so images were augmented with three step augmentation process and further preprocessed by label shuffling, encoding and all images were resized into 96×96 size. Then training dataset were fed into the CNN model and after completing the training, prediction on test set was attained. Then the trained model was deployed on webcam and display connected Raspberry Pi. Now the same model is tested for making final prediction for signs from video streaming of Raspberry Pi. The modeling steps starting from image dataset preparation to final prediction from video streaming is shown in sequential manner in Figure 1.

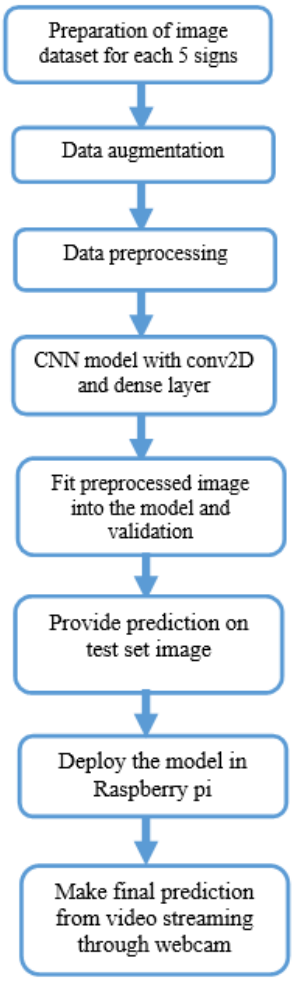

Figure 1. Block diagram of the experimental design.

B. CNN Architecture

The proposed model is based on CNN architecture which proved to be much reliable while working with image dataset [7]. The sequential model used here consists of three convolutional layers and two fully connected dense layers. Rectified Linear Unit (ReLu) activation function is used for all layers except the last dense layer in which SoftMax function is used as activation function. Different layers of CNN extract unique features from images. To minimize the loss function, cross entropy loss is used for compiling the model. It is calculated by,

$$CE = \frac{1}{M}\sum_{p}^{M} -log\left(\frac{e^{s_p}}{\sum_{j}^{C} e^{s_j}}\right) \quad (1)$$

In equation (1), M denotes positive classes of the sample, Sp is the CNN score for each positive class and scaling factor 1/M is used for making the loss invariant to the

number of positive classes. As this is a multi-class classification problem, from C output neurons, CNN will provide a single output for positive class and other classes are treated as negative class.

IV. Execution Process

In this section, the detail execution process is portrayed from dataset making to sign detection through Raspberry Pi. The key points are discussed briefly in following parts of this section.

A. Dataset Preparation

The image dataset is created from custom handmade images following Bangla sign language dictionary [8]. It is divided into two parts, training image set and testing image set individually for each 5 signs. The images are captured by varying backgrounds and brightness conditions. The test image set includes images taken from two separate distances which are 0.5 m and 1 m respectively. Some instances of the training dataset and testing dataset is provided below.

Figure 2. Sample images of 'Count' sign from training set.

Figure 3. Sample images of 'Today' sign from 1 m distance testset

Figure 2 shows samples of images from the training set. A separate test set is prepared for each sign for testing the model accuracy on unseen images. Sample images from the test set of 'Today' sign is shown in Figure 3 which are capture by increasing the distance to 1 meter.

B. Image Augmentation

The training set includes 100 signs from each class which was not sufficient enough to fit in into CNN and proper training [9]. So, the training set is augmented using a 3-step augmentation process. At first the images are flipped horizontally and then by applying gaussian blurring method blurred images are created. The final step was varying the brightness and color channel through adding random brightness coefficient and scaling pixel values in three channels. A glimpse of the training set can be seen from Figure 4. After the completion of 3 step augmentation process, the training set reached to 800 signs per class which enhanced the model to train itself more efficiently.

Figure 4. Sample augmented images of 'Today' sign from training set

C. Image Preprocessing

As training set images had variety of sizes, they were converted into 96×96 resized pixel value. The original aspect ratio was maintained and label files of the images were shuffled to make the training unbiased.

D. Training and Validation in CNN

After augmentation is done the training dataset was comprised of enough signs for each class. Then 30% of the data are kept for testing by splitting the whole training data into training and validation set of 70:30 ratio. In case of setting up values for the model parameters different layer configurations are considered [10]. Those configurations were trained on the augmented image dataset by using specific parameter values shown in Table I.

TABLE I. LAYER PARAMETERS AND VALUES

Name of the Parameter	Value
Number of Epochs	10
Batch Size	16
Input Image Size	96×96
Learning Rate	0.01
Pooling Size	3×3
Verbose	1

E. Testing CNN with Test set images

After completing the training process, the model is tested with validation set images for both 0.5 meter and 1 meter distance datasets. The validation on test dataset showed similar accuracy of 97.54% for 0.5meter dataset and 93.18% for 1meter dataset. A sample of 'Time' sign detection from 0.5 meter distance test set is shown in Figure 5.

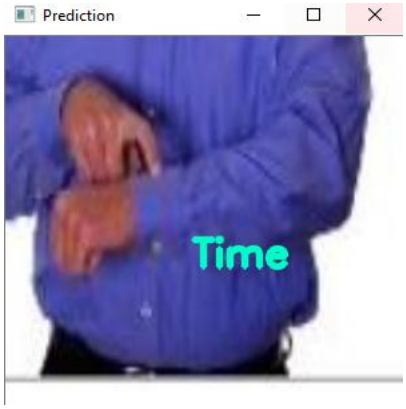

Figure 5. Sample of prediction from test set image of 'Time'

Though the test set images were unknown to the model it predicted with satisfactory accuracy level.

V. HARDWARE IMPLEMENTATION

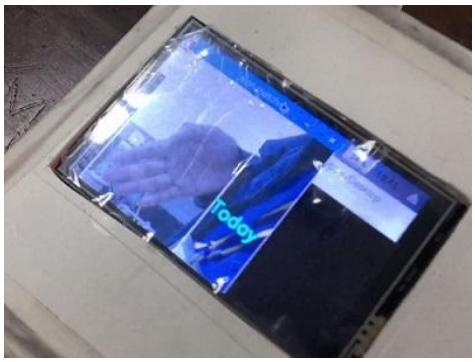

Figure 6. Identification of sign from live video in Raspberry Pi Display

Though the model could identify signs from test set images the next challenge was to deploy the trained model into Raspberry Pi. Before deploying the model with trained weights, Keras, OpenCV and other necessary python libraries were installed into Raspberry pi. For viewing the video output with predicted sign, a 3.5inch display is connected with the Pi. Portable power supply capable of providing 5v and 2.5A and 720p webcam is connected with the GPIO pins of raspberry pi to make it a portable device for translating sings. After assembling all parts correctly, the trained model was then tested for identifying signs from video streaming which is shown in Figure 6.

VI. RESULTS AND DISCUSSION

The choice of encoding method has a great impact on the prediction model. In this case, rather than using popular one hot encoding method, scikit learn label encoding is used. Because label encoding provided less root mean square error (RMSE) than one hot encoding method. As RMSE is less in case of label encoder method so it results in better accuracy which is increased from 86.4% to 97.54%. Table II shows the labeling of five signs by using both encoding styles.

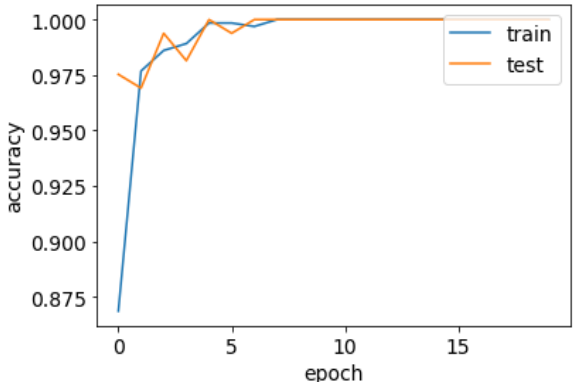

Figure 7. Accuracy vs epoch curve for 0.5m testset

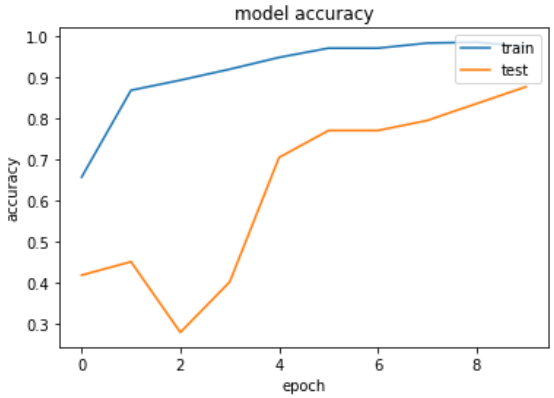

Figure 8. Accuracy vs epoch curve for 1m testset

TABLE II. ENCODING METHODS OF TRAINING DATA

Name of the Sign	Label Encoding	One hot Encoding
Count	0	10000
Time	1	01000
Today	2	00100
Request	3	00010
If	4	00001

It can be clearly seen from Figure 7 and Figure 8 that, with 0.5 meter testset the accuracy vs epoch reaches 97.54% within 10 epochs and for 1 meter distance testset the accuracy was 93.18%. Though the accuracy vary with distances for images but in case of identifying signs from videos it shows approximately 94% accuracy where frame rate was 30fps. On top of that, it took around 1 to 1.5 seconds for identifying each signs from the video which is very efficient in terms of feasibility.

VII. Conclusion And Future Scopes

The proposed method offers an efficient resolution for conventional sign language learning methods. In these approaches, huge training data are required and detection of signs from video was also very time consuming. By using customized ROI segmentation method, the model no longer needs to go through computationally heavy workflow of localizing the hand area by its own. The user of the device can move the preloaded bounding box on screen to the hand area of deaf person and thus only the area inside the bounding box is sent to the CNN model for prediction. Integrating the model with ARM cortexA53 embedded Raspberry pi adds flexibility and portability facility to the device. As a result, the accuracy level as well as detection speed is increased. Because of the unavailability of the dataset on Bangla sign language only 5 signs are used for making this translator device. In future, more signs will be added to the device and a graphical user interface (GUI) will be introduced along with the existing model for enhancing its operation.

References

[1] S. Ahmed, R. Islam, M. R. Zishan, M. R. Hasan, and M. Islam, "Electronic speaking system for speech impaired people: Speak up," *2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, 2015.

[2] Y. Madhuri, G. Anitha, and M. Anburajan, "Vision-based sign language translation device," *2013 International Conference on Information Communication and Embedded Systems (ICICES)*, 2013.

[3] Y Ji, S Kim, KB Lee, "Sign Language Learning System with Image Sampling and Convolutional Neural Network." Robotic Computing (IRC),IEEE 2017.

[4] Sahoo, Ashok K. Gouri Sankar Mishra, and Kiran Kumar Ravulakollu,"Sign language recognition: state of the art," ARPN Journal ofEngineering and Applied Sciences, vol. 9. Feb. 2014, pp. 116-134.

[5] Jens Forster, Oscar Koller, Christian Oberd¨orfer, Yannick Gweth, Hermann Ney, "Improving continuous sign language recognition: Speech recognition techniques and system design," Workshop on Speech and Language Processing for Assistive Technologies, Grenoble, France, Aug.2013.

[6] Sampada S. Wazalwar & Urmila Shrawankar (2017) Interpretation of sign language into English using NLP techniques, Journal of Information and Optimization Sciences, 38:6, 895-910

[7] Misra, Ishan, C. Lawrence Zitnick, and Martial Hebert, "Unsupervised Learning using Sequential Verification for Action Recognition,"arXiv preprint arXiv:1603-08561, Mar. 2016.

[8] M. D. S. Rahman, K. Arnesen, and M. D. A. Hossain, *Bangla Sign Language Dictionary*. Dhaka: National Centre For Special Education, Ministry of Social Welfare, 1994.

[9] H. Zunair, N. Mohammed, and S. Momen, "Unconventional Wisdom: A New Transfer Learning Approach Applied to Bengali Numeral Classification," *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, 2018.

[10] Simard, Patrice Y. David Steinkraus, and John C. Platt, "Best Practicesfor Convolutional Neural Networks Applied to Visual Document Analysis," ICDAR. Vol, Mar. 2003.