Proceedings of ICAEEC-2019, IIIT Allahabad India, 31st May - 1st June,2019

Hardware Implementation of Sign Language to Text Converter using Deep Neural Networks

Guda Harsha Vardhan[1], G.Sri Venkat[1], M.G.Pratyusha[1],
Kunal Gupta[1], Volam Priyanka[1], and P V Sudeep[1]

Department of Electronics and Communication Engineering,
National Institute of Technology Calicut, Kerala, INDIA.

harshavardhan.guda@gmail.com srivenkat1000@gmail.com
mgpratyusha@gmail.com guptakunal853@gmail.com
priyankavolam@gmail.com sudeep.pv@nitc.ac.in

Abstract: The exchange of information performed in different ways such as speech, signals, or writing is called communication. Communication in deaf and mute is performed with the use of sign languages; however, they face many difficulties at the time of interacting with the common man and computers. Research is needed in this field to bring deaf-mutes more into the light of society and to increase their interaction with the common man. This paper aims at developing a system that uses Convolutional Neural Networks (CNN) and Faster R-CNN to extract the patterns in the feature vectors of each sign of twenty-six alphabets and further uses the results to recognize those signs. Different methods for the design of neural networks are proposed in this paper to enhance the performance. A standalone device is developed by implementing the pre-trained modified VGGNet models on Pi 3 A+ device.

Keywords: Artificial Neural Network, Hand gesture recognition, Motion analysis, Raspberry Pi, Sign Languages, VGGNet.

1 Introduction

Sign Languages are mainly used by the deaf to interact with others. They express their language by some particular manual actions. These are the languages with their own signs and meanings which vary from place to place. There are different sign languages such as American [22], Indian [24], Arabic [16], and International sign languages [23]. Fig. 1 shows the ISL sign chart of 26 English alphabets.

Recognition of gestures is the primary thing to do in sign to text conversion. There are many methods like glove detection method [1], Vision based detection [15], Scale Invariant Feature Transform (SIFT) algorithms [2], Artificial neural networks [6] [7] [30] to detect the sign and convert them to text.

In the glove detection method [1], the glove consists of sensors that collect the information and uses it to detect the signs. It normally uses pressure sensors to detect the pressure between fingers and process the data to detect the gestures. Vision-based approach [15] for classifying signs [29] is another popular technique. In another approach, American Sign Language alphabets are recognized using SIFT algorithms [2].

Indian Sign Language has adopted signs [24] from both American [22] and British [25] Sign Languages. It consists of both single and double hand sign representations. So it is slightly difficult to detect and convert them to text than its American counterpart. Sakshi *et al.* [4] proposed a method which uses SIFT to detect key points in the image, a database of key points of all the alphabets is made. A new image's key points are compared with those ones on the database to get the label of the most accurate and closest match. Joyeeth *et al.* [5] used an approach to represent the images in the form of vectors and calculate eigenvectors of covariance matrix which is obtained from the image vector. Eigen vector of the new image is used to calculate Euclidean distance with the already existing ones in the training set to get the label of the new image.

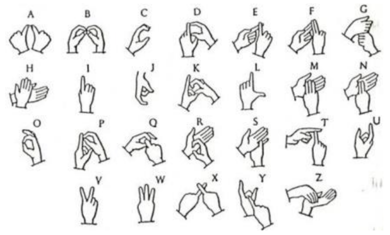


Fig.1: ISL chart representing 26 English alphabets.

Due to the recent developments in Artificial Intelligence and the ease of classifying objects using Convolutional Neural Networks (CNN) [12], researchers are now resorted to CNN for image recognition and converting them into text. CNN is a class of deep neural networks which is mainly used for image classification, developed by taking visual cortex as reference. The hidden layers mainly consist of Convolutional layers, RELU activation functions, pooling, fully connected and normalization layers.

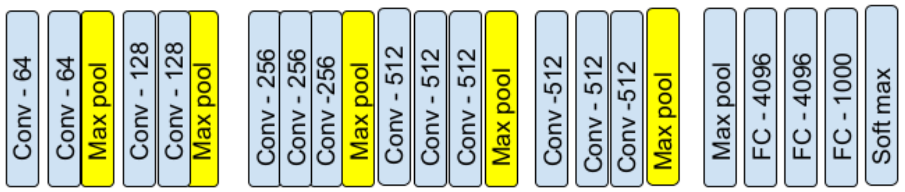


Fig.2: Layers in VGGNet

Although the neural networks are used in works carried out by Padmavati *et al.*[6] for training. The models are trained with datasets comprised of only singlehanded images. In addition, it used a dataset of numbers, the angle between fingers as feature vectors. In another approach, by Sajanraj *et al.*[7] a real-time system uses skin segmentation to find the segmented region and it is fed to the neural network model that is developed to predict the sign, nonetheless the method has been intended for single-handed signs of numbers. P.Kishore *et al.*[30] developed a real-time sign language recognition system that recognizes signs from videos. It uses active contour segmentation to segment and tracks the hand and image classification is done by an artificial neural network. However, it is not suitable for implementation on small scale computers such as Raspberry Pi.

Most of the ISL recognition methods, which have been developed so far use single hand signs nonetheless ISL uses both hands for representations[8][9]. Majority of them cannot be employed to small embedded systems like Raspberry pi because of compatibility issues and storage limitations.

The main objective of this paper is to develop a system that can recognize all ISL alphabets and uses faster R-CNN with the base network as modified VGGNet for more accurate results. The modified VGGNet is trained and the model is implemented on Raspberry Pi for real-time recognition of signs in ISL.

The rest of the paper is structured as follows. Section 2 specifies the software and hardware used. Section 3 provides an insight into the methodology employed. The test results and related discussions are provided in Section 4. The paper is concluded in Section 5.

2 Background

In this section, we briefly describe the software tools and the hardware used for the implementation of the ISL to text conversion system. Also, we discuss different quality metrics used in this paper to evaluate the proposed method.

Python is used as a programming language to develop the proposed system. This essentially requires installation of Keras [27] and OpenCV [28] libraries. Keras is a framework for development and deployment of neural networks. Keras is written in python which is an open source neural network library and has capabilities to run tensor flow as back-end. OpenCV is an open sourced library for carrying out computer vision operations. OpenCV is used for pre-processing the images taken.

Raspberry Pi [26] is used to deploy the developed system. It is a mini-computer with a size of a debit card. It has an ARM Cortex-A53 64-bit CPU working at a clock frequency of 1.4GHz with a RAM of 512MB. It has a provision to connect a camera directly to its port given on the board. A pi camera of 5MP is used to capture real-time images. Raspberry Pi is chosen as for its most active, community support, small size.

VNC Server which is run on Raspberry Pi and VNC Viewer which is run on the local computer is used to get the display of Raspberry Pi on to the computer.

Evaluation of network is carried out using precision, recall [13], f1-score [13]. Precision is defined as the ratio of positive identifications that are correct as shown in equation 1. Recall is defined as ratio of actual positives that are correct as shown in equation 2.

$$precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (1)$$

$$recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (2)$$

$$f_1 - score = \frac{2 \times precision \times recall}{precision + recall} \quad (3)$$

3 Methodology

A neural network has to be explored in order to recognize a given sign. There are neural networks developed for object recognition with good accuracy values. One of it can be taken and adopted for the purpose of sign recognition. VGGNet(VGG stands for Visual Geometry Group)[3] is one of the CNN architecture that has 16 Convolutional layers and the preferred option for extracting features from the images as it has a lot of filters. The layers in the VGGNet are as shown in Fig.2. Input size of the network model is 224 × 224 × 3 i.e., a color image of size 224 × 224, it has 16 layers.

VGGNet is used in this work as it has the highest top-1 and top-5 accuracy next to Resnet-50 [10]. Here top-1 accuracy is the case where the correctly predicted class has the highest accuracy, top-5 accuracy is the case where the correctly predicted class lies in any of the 5 highest probabilities. Nonetheless resnet-50 is deep with many layers; hence VGGNet is chosen and modified as to make the model size small. It is necessary to make the model size small as it has to be deployed on Raspberry Pi which has computational constraints and to decrease the time taken for predictions. Thus make it work in real-time.

3.1 Modified VGGNet

As discussed earlier in this section, we aim at a fast and reliable ISL to text conversion system with better accuracy. If the number of neurons in each of the layers of the deep neural network is reduced, it reduces computation complexity and hence, provides faster output results. It obviously reduces accuracy. Hence a trade-off between accuracy and computation time is required. This left an option to modify already developed CNN named VGGNet by decreasing each convolution layer's size appropriately to have a good trade-off between accuracy and computation time. The modified VGGNet has an input size of 64 × 64 × 3 i.e., a color image of size 64 × 64. This, in turn, reduced the parameters by 30 folds as shown below in the Table 1.

Keras is used to build the modified VGGNet and is trained using Keras in python. Fig. 3 shows layers of Modified VGGNet CNN which has been discussed so far.

	VGGNet	Modified VGGNet
Trainable parameters	138,357,544	4,623,811
Non-trainable parameters	0	2,112

Total parameters	138,357,544	4,625,923

Table 1: Table showing the parameters in the VGGNet and modified VGGNet.

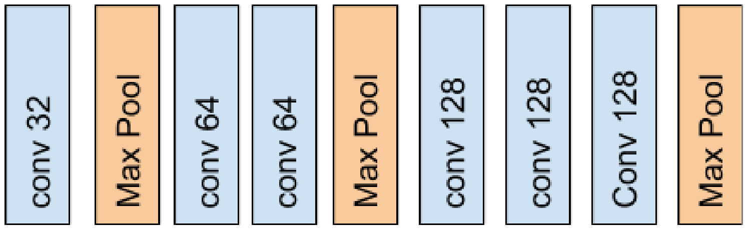


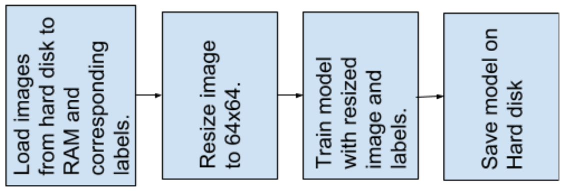
Fig.3: Modified VGGNet model.

3.2 Faster R-CNN

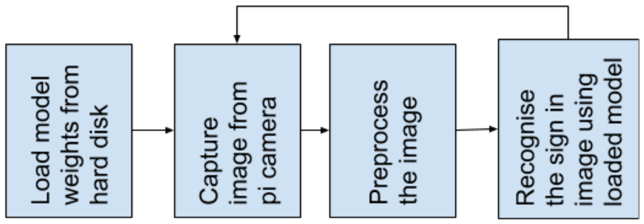
Region-based Convolutional Neural Networks(R-CNN) [13] are used in the application where the image which has to be recognized contains multiple classes of recognizable objects and to draw a bounding box over the recognized object. The proposed system needs to recognize a sign from the image under varying conditions and also when an image consists of other objects/signs in the image. Thus a Faster R-CNN is chosen to implement the proposed system. Faster R-CNN [14] has Region Proposal Network (RPN) and a network which uses the proposed regions from RPN to recognize objects. The output of an RPN could be a bunch of boxes or proposals which are further passed through a classifier and a regressor to check about the ubiquity of the objects present in the image. To achieve more accuracy in this process, RPN also predicts the likelihood associated with the anchor whether it is background or foreground in the image and modifies the anchor accordingly. After RPN, we have a tendency to get projected regions with totally different sizes.

It is difficult to create and associate an economical structure to figure on options with totally different sizes. To overcome this issue Region of Interest Pooling (ROI) is used instead of Max-Pooling which incorporates a fixed size. The input feature map is splitted a hard and fast variety by ROI pooling which has on an average equal sizes and Max-pooling is applied on features extracted previously from each region. This makes output size same keeping it independent from the input size by the use of ROI pooling.

Modified VGGNet is used as base network; the layers are modified appropriately to reduce the parameters to train on the dataset. The dataset is labeled to get the boxes which bound the signs; it is used to train RPN. Training is done on the Google Cloud Platform with a GPU of 16 GB NVIDIA Tesla P-100. Fig. 4 shows training and real-time prediction processes.


Load images from hard disk to RAM and corresponding labels.
Resize image to 64x64.
Train model with resized image and labels.
Save model on Hard disk

a. Training process.


Load model weights from hard disk
Capture image from pi camera
Preprocess the image
Recognise the sign in image using loaded model

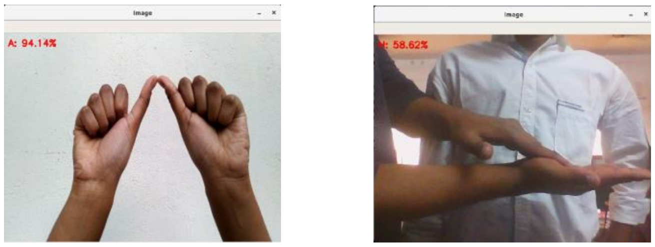
b. Prediction in real-time.

Fig.4: Flow diagrams of training and prediction.

4 Results and Discussions

People in different regions of India use different signs for communication. Most commonly used signs are collected and a data-set is prepared for those signs. The signs used in this paper contain the use of both the hands [7] rather than single hand considered in research work carried out by Padmavati *et al.* [6] [7]. A dataset of signs of twenty-six alphabets posed by two male and two female signers under different lighting, backgrounds are taken as different video footages and frames are extracted. Each alphabet sign has an average of 1,000 images, these totals up to 28,000 images for whole alphabets' signs.

The modified models are trained with a dataset collected. The modified VGGNet CNN is trained on NVIDIA GPU GeForce GTX 745 of 4GB Memory with a batch size of thirty-two and seventy-five epochs. Categorical Cross-entropy is used for loss function. After training on whole dataset an accuracy of 93.39 % is achieved with a loss of 0.0856. The trained model is saved on a hard disk for future use to predict signs from images.



a. correct prediction of sign 'A'. b. Falsely predicted sign 'Z' as 'H'.

Fig.5: Example results using Modified VGGNet CNN.

It is implemented on Raspberry Pi 3 A+ the pi camera is used to take images as input in real time. Images are captured at a rate of one image per 1.5 seconds as the Pi took an average time of 1.5 seconds to predict a given image.

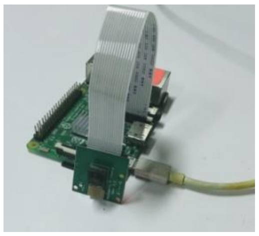
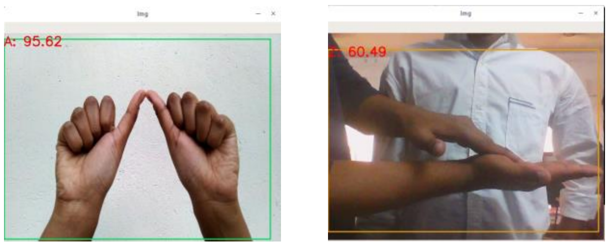


Fig.6: Hardware implementation on Raspberry Pi

Nonetheless the conditions such as light, background varied the predicted results deviated a lot from the original results. Fig. 5 shows the predictions of modified VGGNet CNN model on some images taken in real-time. A sign with alphabet 'A' is given to the system and it was predicted correctly with confidence of 94.14% as shown in Fig.5.

In the experiments performed, it is found that for images taken in good lighting conditions the system showed satisfactory results.

A sign consisting of alphabet 'Z' is given to the system, Fig. 5b shows that the modified VGGNet CNN failed to detect the given sign as 'Z' instead predicts it as 'H'. It failed to detect as it has a lot of background in the image and it considers the whole image as a single sign, tries to predict nonetheless that is not the case for that image. That is the reason for approaching the recognition problem with the R-CNN.



a. Correct prediction of sign ’A’ b. Correct prediction of ’Z’

Fig.7: Example results using Modified VGGNet R-CNN.

To get good prediction results, the method of R-CNN is applied as mentioned in the earlier section. It shows improved results over the modified VGGNet CNN model and the major advantage is that the R-CNN model draws bounding boxes over the sign in the input image and it can also detect multiple signs in a single image and draws bounding boxes over them. The Same sign of ’A’ from Fig.5a is given to modified VGGNet R-CNN it can be observed that accuracy has increased to 95.62%. The same sign of alphabet ’Z’ as in Fig.5b is given to the system nonetheless it has predicted it correctly as ’Z’ as shown in Fig. 7b.

Table 2 gives a comparison between modified VGGNet CNN and R-CNN proposed in this paper in terms of precision, recall and f1-score. The drawback of the proposed ISL to text converter is that it takes more time to predict the sign in the image when compared to Modified VGGNet CNN. On average, it takes up to 4 seconds on Raspberry Pi 3 A+. Fig. 6 shows the hardware implementation of obtained models on Raspberry Pi.

Model	Average	Precision	recall	f1-score
Modified VGGNet CNN	micro-avg	0.91	0.90	0.90
	macro-avg	0.91	0.89	0.90
Modified VGGNet R-CNN	micro-avg	0.93	0.91	0.92
	macro-avg	0.92	0.94	0.93

Table 2: Performance comparison of different models using precision, recall and f1-score.

References

1. Lopez-Noriega, J. E., Ferna´ndez-Valladares, M. I., & Uc-Cetina, V. (2014, September). Glove-based sign language recognition solution to assist communication for deaf users. In Electrical Engineering, Computing Science and Automatic Control (CCE), 2014 11th International Conference on (pp. 1-6). IEEE.
2. Nachamai. M1,”Alphabet Recognition of American Sign Language: A Hand Gesture Recognition Approach Using SIFT Algorithm”.
3. Simonyan, Karen, and Andrew Zisserman. ”Very deep convolutional networks forlarge-scale image recognition.” arXiv preprint arXiv: 1409.1556 (2014).
4. Sakshi Goyal, Ishitha Sharma, S. S. Sign language recognition system for deaf and dumb people. International Journal of Engineering Research Technology 2, 4 (April 2013).

5. Joyeeth Singh, K. D. Indian sign language recognition using eigenvalue weightedeuclidean distance based classification technique. International Journal of Advanced Computer Science and Applications 4, 2 (2013).
6. Padmavathi. S, Saipreethy.M.S, V. Indian sign language character recognition using neural networks. IJCA Special Issue on Recent Trends in Pattern Recognition and Image Analysis, RTPRIA (2013).
7. Sajanraj, T. D., and Mv Beena. ”Indian sign language numeral recognition using the region of interest Convolutional neural network.” 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT). IEEE, 2018.
8. Adithya, V., P. R. Vinod, and Usha Gopalakrishnan. ”Artificial neural network based method for Indian sign language recognition.” 2013 IEEE Conference on Information & Communication Technologies. IEEE, 2013.
9. Ghotkar, Archana S., and Gajanan K. Kharate. ”Study of vision based hand gesture recognition using Indian sign language.” Computer 55 (2014): 56.
10. He, Kaiming, et al.”Deep residual learning for image recognition.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
11. Howard, Andrew G., et al.”Mobile nets: Efficient Convolutional neural networks for mobile vision applications.” arXiv preprint arXiv: 1704.04861 (2017).
12. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. ”Imagenet classification with deep Convolutional neural networks.” Advances in neural information processing systems. 2012.
13. Girshick, Ross, et al. ”Rich feature hierarchies for accurate object detection and semantic segmentation.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.
14. Ren, Shaoqing, et al.”Faster r-cnn: Towards real-time object detection with region proposal networks.” Advances in neural information processing systems. 2015.
15. Wu, Ying, and Thomas S. Huang. ”Vision-based gesture recognition: A review.”International Gesture Workshop. Springer, Berlin, Heidelberg, 1999.
16. Mohandes, Mohamed, Mohamed Deriche, and Junzhao Liu. ”Image-based and sensor-based approaches to Arabic sign language recognition.” IEEE transactions on human-machine systems 44.4 (2014): 551-557.
17. Mohandes, Mohamed, Junzhao Liu, and Mohamed Deriche. ”A survey of image based arabic sign language recognition.” 2014 IEEE 11th International MultiConference on Systems, Signals & Devices (SSD14). IEEE, 2014.
18. Khan, Rafiqul Zaman, and Noor Adnan Ibraheem. ”Hand gesture recognition: aliterature review.” International journal of artificial Intelligence & Applications 3.4 (2012): 161.
19. Al-Ahdal, M. Ebrahim, and Md Tahir Nooritawati. ”Review in sign language recognition systems.” 2012 IEEE Symposium on Computers & Informatics (ISCI). IEEE, 2012.
20. Cheok, Ming Jin, Zaid Omar, and Mohamed Hisham Jaward. ”A review of handgesture and sign language recognition techniques.” International Journal of Machine Learning and Cybernetics 10.1 (2019): 131-153.
21. Wu, Ying, and Thomas S. Huang. ”Hand modelling, analysis and recognition.”IEEE Signal Processing Magazine 18.3 (2001): 51-60.
22. Valli, Clayton, and Ceil Lucas. Linguistics of American Sign Language: an introduction. Gallaudet University Press, 2000.
23. Supalla, Ted, and Rebecca Webb. *The grammar of International Sign: A new look at pidgin languages.* Language, gesture, and space (1995): 333-352.
24. Tomkins, William. *Indian sign language.* Vol. 92. Courier Corporation, 1969.

25. Sutton-Spence, Rachel, and Bencie Woll. The linguistics of British Sign Language: an introduction. Cambridge University Press, 1999.
26. Raspberry Pi Organization. https://www.raspberrypi.org/products/ raspberry-pi-3-model-a-plus/
27. Chollet, F. (2015) keras, GitHub. https://github.com/fchollet/keras
28. Itseez, Open Source ComputerVision Library, 2015. urlhttps://github.com/itseez/opencv
29. Rautaray, Siddharth S., and Anupam Agrawal. ”Vision based hand gesture recognition for human computer interaction: a survey.” Artificial intelligence review 43.1 (2015): 1-54.
30. Kishore, P. V. V., and P. Rajesh Kumar. ”Segment, Track, Extract, Recognize and Convert Sign Language Videos to Voice/Text.”