Real-Time Sign Language Character Recognition

Zubin Bhuyan
Student id: 01744486
Department of Computer Science
zubin_bhuyan@student.uml.edu

***Abstract*—The last decade has seen the domain of hand gesture recognition and sign language translation mature into a research area of great potential with newer challenges. This is an important area of research which can have impact in numerous fields ranging from HCI to computer vision. The domain of SLR is quite broad and this project focuses on the task of detecting sign language spelling, i.e. identifying hand signs for letters of the English alphabet. A MobileNet SSD, detect hands in various configurations from video frames and forwards its result to another CNN which maps these hands to letters of the English alphabet. This model is capable of detecting signs in realtime. The model showed an accuracy of 92% when tested with the MNIST sign language dataset.**

I. Introduction

Sign language is a sequence of hand movements and gestures which is used to convey messages, and serves as a means of communication for people with speaking and hearing disabilities. Most sign languages use not just hands to convey messages; facial expressions and posture can also convey important contextual information. Sign languages are natural languages, and so for any software which attempts to interpret sign language faces the challenges of NLP as well as the challenges of visual analysis.

American Sign Language (ASL) is one of the hundreds of sign languages used all over the world. In 2006, it was estimated that ASL is the primary language for more than 350,000 people [1]. The focus of this project is to develop a deep neural solution to recognize character signs of ASL.

The remainder of the paper is divided in four more sections. Section II talks about the motivation for research in the area of sign language recognition. Section III lists a few recent work in this area. The datasets used and the proposed approach is described in section IV. Experimental results are discussed in section V, which is followed by the conclusion.

II. Motivation

Sign language recognition (SLR) systems can be of great help in overcoming communication hurdles for people with listening and hearing disabilities. Such systems can be the *lingua franca* for interaction with those who do not know sign language. Automatic translation of audio and videos is another possible application of such softwares. SLR tools can also be used as an alternative to voice input for deaf users. Additionally, such research will further facilitate the study of sign language linguistics and enable automated annotation and meta-data generation.

It is estimated that spelling characters comprises of 12% to 35% of ASL [2]. Moreover, hand spelling recognition systems can be thought of as the first step towards designing a more generic sign recognition system. The implementation of the model for this project can detect more than one hand in an image; this can further be used to recognize more complex signs which involves both hands.

III. Related Work

SLR is generally categorized as a type of gesture recognition. Two common approaches for hand sign and gesture recognition are: *i.* Vision-based, and *ii.* Sensor based. Vision based approaches involve capturing visual data of the hand using single, stereo and/or depth cameras, and color markers such as colored gloves, LEDs and special props. Devices such as Kinect and Leap Motion [3] have also been used. Sensor based approaches use devices to measure acceleration and orientation of fingers and hand, EMG and other measurable values. Recent advances in the area have been made by employing vision based approaches.

A review of this area done about two decades ago [4] show that most work being done was limited to static hand postures. HMM was the most popular choice of researchers. Starnet, et. al in [5] show how HMM was successfully used for sign recognition in real-time. Time delay neural networks were also used to capture the temporal aspect of hand movement. More recent reviews [6], [7] show how advances in deep learning and computation technology has dramatically improved SLR and hand gesture recognition in general.

Since CNNs are usually good for image recognition, they are popular for hand gesture recognition too. The input to such systems can be simple images or 2.5D/3D images. A recent work [8] uses CNNs in conjuction with LSTM for sign language finger spelling in videos from websites. This work is especially interesting because the videos had different quality and frame rate.

IV. Proposed Approach

The approach proposed in this project attempts to solve the problem of sign language character recognition in the following two steps:

1) From the video frame, identify hands and determine the bounding boxes for them.
2) Analyze the hands inside the bounding boxes for known hand signs.

One part of the proposed architecture recognizes only hands, and the other part to recognize hand signs. Two different datasets were used for the training. Subsection IV-A lists the details of the datasets and describes why they are appropriate for the respective stages. Subsections IV-B and IV-C describe the two components of the architecture proposed.

A. Datasets

1) EgoHands dataset [9]: This is a freely available dataset[1] of annoted images of hands of people performing various tasks. These images are taken from videos recorded by google images. There are a total of 15,053 ground truth labeled hands (pixel-level segmentation).

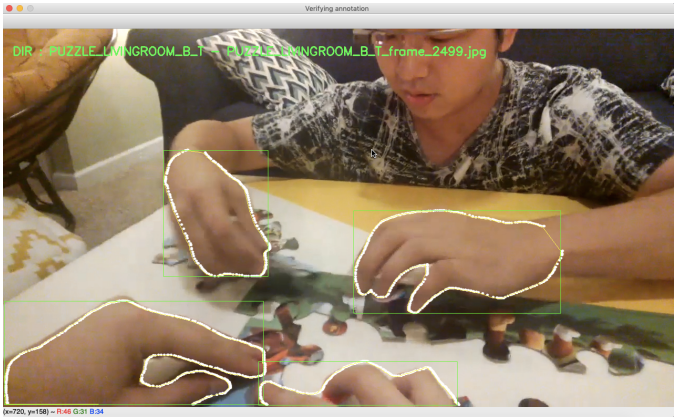

Fig. 1. An image of the EgoHand dataset.

Fig 1 is an example of an image of the dataset. The white points marking the hands are in a separate file. A python script was used to visualize this image with the white boundary and a green bounding box. For training only one class, "hand" was used; instead of user's left and right, and social partner's left and right.

2) Sign Language MNIST dataset : This dataset [2] has gray scale images of size 28x28 for 24 hand sign characters, i.e. 24 characters of the alphabet (Letters j and z are not included as they involve movement). There are 27,455 train cases and 71,72 test cases, all in labelled CSV format.

B. Detecting hands with MobileNet v1 and SSD

Transfer Learning involves reusing a model developed for a task as the starting point for a different task. For the task of identifying hands in a video frame or image, we used the *ssd_mobilenet_v1_coco* model from the Tensorflow detection

[1]http://vision.soic.indiana.edu/projects/egohands

[2]www.kaggle.com/datamunge/sign-language-mnist

Fig. 2. Sample handsigns from the sign language MNIST dataset.

model zoo[3]. (This model is pretrained on the COCO dataset.)[4] This model is also augmented with the single-shot multibox detection technique (which was originally used with VGG) [10].

Google's MobileNetV1[5] [11] makes use of Depthwise Separable Convolution[6] and point-wise convolutions along with a *width multiplier* and a *resolution multiplier* results in a simpler to train network, and a relatively smaller model which can perform faster. As shown in fig 3, depthwise convolution is performed spatially on each channel and is followed by pointwise convolution to adjust to the required dimension. Fig 4 shows the architecture of the model.

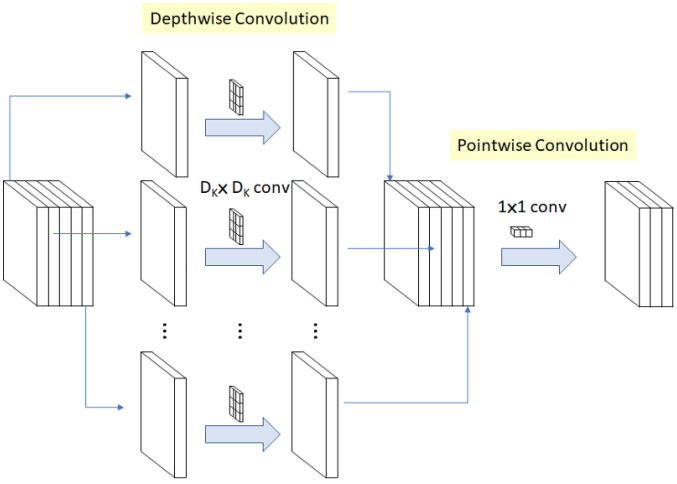

Fig. 3. Sample handsigns from the sign language MNIST dataset.

[3]https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md

[4]Two similar and easy to follow tutorial on this topic can be found at: https://medium.com/@victor.dibia/how-to-build-a-real-time -hand-detector-using-neural-networks-ssd-on-tensorflow -d6bac0e4b2ce and https://github.com/jkjung-avt/hand-detection-tutorial

[5]https://ai.googleblog.com/2017/06/mobilenets-open-source-models-for.html

[6]An easy to understand explanation of MobileNet can be found here https://bit.ly/2V2Vfq6

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Fig. 4. MobileNet Architecture

C. Identifying hand sign

The bounding boxes of detected hands in the previous step are forwarded to this CNN which checks for known hand signs. This network is much smaller than the previous one and has only two convolution layers and two max-pool layers. Batch normalization is applied to the output of both convolution layer. The filter sizes are 3x3 and stride is 1; the max-polling stride was 2, and padding is not applied. This is followed by two fully connected layers at the end with soft-max classifying the input to one of the 24 letters of the alphabet. The batch size chosen was 5, learning rate was 0.001 and momentum was 0.2.

V. Experimental Results

The training was done on a desktop with an NVIDIA GeForce GTX 1070 running Ubuntu 16.04. The training was done separately for MobileNetv1 model and the CNN with the EgoHands and MNIST sign language datasets respectively.

Fig 5 shows how the loss changes with steps. After 200,000 steps, the loss was at 2.57. Table I shows the average precision for different Intersection over Union (IoU).

TABLE I
Average Precision at variour IoU

AP @ 0.5 IoU	0.968
AP @ 0.75 IoU	0.813
AP @ 0.5:0.95 IoU	0.678

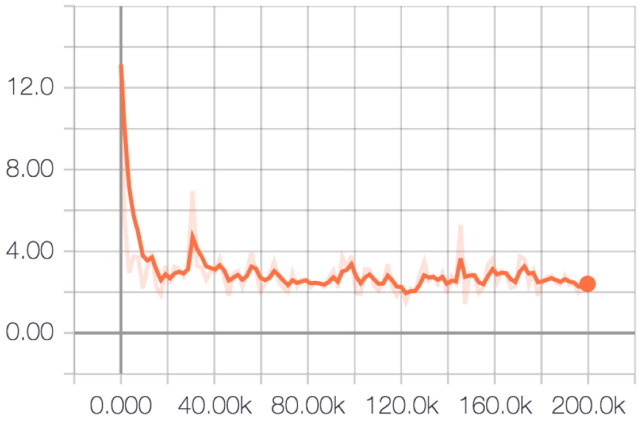

Fig. 5. Traing MobileNetV1SSD with EgoHand.

The loss for training and validation sets for the hand sign dataset is shown in fig 6. The accuracy achieved on the test set was 92% and is shown in fig 7.

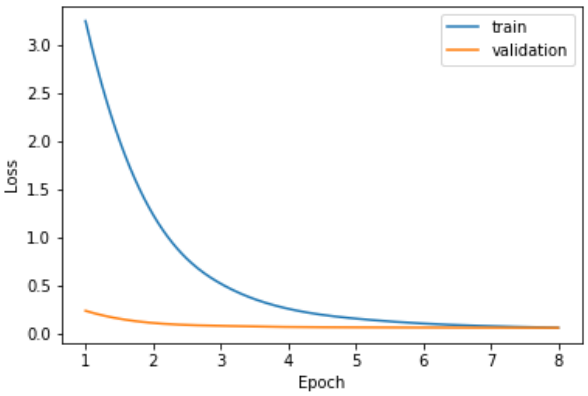

Fig. 6. Traing CNN with MNIST hand sign dataset.

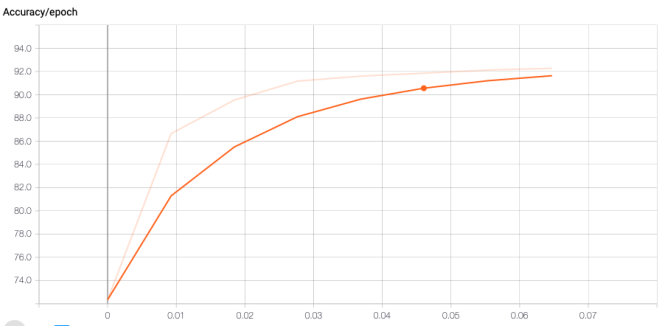

Fig. 7. Accuracy of CNN on MNIST hand sign test cases.

For real-time detection, Open CV for Python was used. The detection could be done with about 18 fps on the desktop with the GTX 1070. On a 2017 MacBook Air, the fps dropped drastically down to less than 8. Fig 8, 9 are screenshots of realtime hand sign detection.

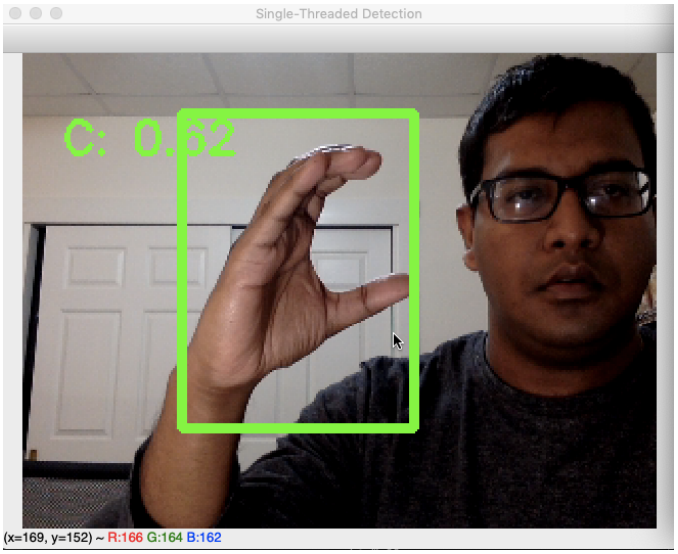

Fig. 8. Screenshot of realtime hand sign detection.

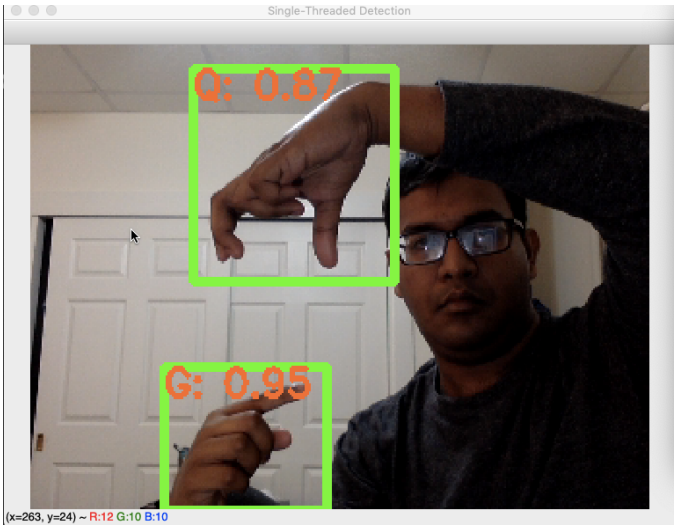

Fig. 9. Realtime hand sign detection with multiple hands.

VI. Conclusion

In this project, real time hand sign detection was done for American Sign Language characters using a two step detection process which employed Google's MobileNetV1 and a CNN. The implemented system could process about 8 frames per second on a MacBook Air (without graphics card) in real time. Further improvements can be achieved by training the model with images of hand-signs in different lighting condition.

This work can be extended for detection of signed sentences which involve movement of hands. Since there are models which can detect poses and gestures, and it will be interesting to see they can be further trained with a *small number* of hand signs with movement. There is also scope for improvement by training on generic videos. However, such videos are rarely tagged, so annotation is another task which has to be performed before they are used for training. Another possible area, at times, require both hands to convey a message. Extending the current work to detect two handed signing seems like a rewarding path to follow.

Furthermore there is scope for research in automatic annotation and other linguistic analysis such as machine translation and analysis of feasibility of using text embeddings for sign languages.

References

[1] R. Mitchell, T. Young, B. Bachleda, and M. Karchmer, "How many people use asl in the united states? why estimates need updating," vol. 6, no. 3, pp. 306–335, 2006.

[2] C. Padden and D. Gunsauls, "How the alphabet came to be used in a sign language," vol. 4, no. 1, pp. 10–33, 2003.

[3] L. E. Potter, J. Araullo, and L. Carter, "The leap motion controller: A view on sign language," in *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*, OzCHI '13, (New York, NY, USA), pp. 175–178, ACM, 2013.

[4] Y. Wu and T. S. Huang, "Vision-based gesture recognition: A review," in *Proceedings of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction*, GW '99, (London, UK, UK), pp. 103–115, Springer-Verlag, 1999.

[5] T. Starner and A. Pentland, "Real-time american sign language recognition from video using hidden markov models," in *Proceedings of International Symposium on Computer Vision - ISCV*, pp. 265–270, Nov 1995.

[6] Suharjito, R. Anderson, F. Wiryana, M. C. Ariesta, and G. P. Kusuma, "Sign language recognition application systems for deaf-mute people: A review based on input-process-output," *Procedia Computer Science*, vol. 116, pp. 441 – 448, 2017. Discovery and innovation of computer science technology in artificial intelligence era: The 2nd International Conference on Computer Science and Computational Intelligence (ICCSCI 2017).

[7] M. J. Cheok, Z. Omar, and M. H. Jaward, "A review of hand gesture and sign language recognition techniques," *Int. J. Machine Learning & Cybernetics*, vol. 10, no. 1, pp. 131–153, 2019.

[8] B. Shi, A. Martinez Del Rio, J. Keane, J. Michaux, D. Brentari, G. Shakhnarovich, and K. Livescu, "American sign language fingerspelling recognition in the wild," pp. 145–152, 12 2018.

[9] S. Bambach, S. Lee, D. J. Crandall, and C. Yu, "Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," *CoRR*, vol. abs/1512.02325, 2015.

[11] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017.