American Sign Language Recognition Leveraging Machine Learning

Abdullah Al Maruf [1]

Abstract

Knowing sign language is important for communicating with deaf people. The majority of people in society do not understand sign language, which causes difficulties for deaf people in everyday life. Sign language can be made easier to understand with the help of modern technology. Machine learning may assist in the recognition of gestures and the translation of such gestures into text or speech. This project trains a model utilizing the pre-trained MobileNetV2 model. We got 0.994 accuracy and 0.0227 loss on the dataset that the model never seen. We also developed a user interface to convert sign language to text in real time.

1. Introduction

American Sign Language (ASL) is a complete, natural language with the same linguistic properties as spoken languages, with grammar that differs from English. ASL is expressed by movements of the hands and face. It is the primary language of many North Americans who are deaf and hard of hearing and is used by many hearing people as well. The number of ASL users in the United States is estimated to be between 250,000 and 500,000 people, including a number of children of deaf adults (Mitchell et al., 2006).

This project aims to automate the recognition of signs and gestures using machine learning.

2. Related Work

American sign language recognition is a well known research sector and researchers are doing their work on this sector for a long time.

(Rivas et al., 2019) demonstrated a five-layer unsupervised encoder-decoder neural model in their work. They used a dataset of segmented images captured with a depth-sensor camera for different subjects. The average accuracy obtained was of 98.9

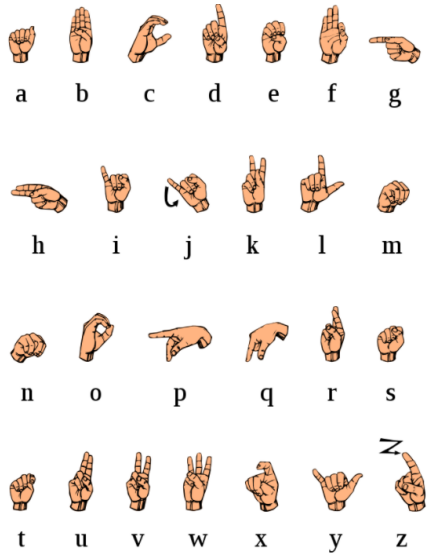

Figure 1. ASL finger-spelling alphabet (also referred to as the American manual alphabet)

(Zafrulla et al., 2011) worked on word and phrase sign recognition using the Kinect. They proposed a new multimodal kinect system and compare results to the CopyCat system which uses colored gloves and embedded accelerometers. They collected a total of 1000 American Sign Language (ASL) phrases using both the Kinect system and the CopyCat sensor platform. On adult data, the Kinect system resulted in 51.5

In the real-time recognition section, the work of (Garcia & Viesca, 2016) was based on convolutional neural network. We utilize a pre-trained GoogLeNet architecture trained on the ILSVRC2012 dataset, as well as the Surrey University and Massey University ASL datasets in order to apply transfer learning to this task. They produced a robust model that consistently classifies letters a-e correctly with first-time users and another that correctly classifies letters a-k in a majority of cases. They achieved 0.9782 validation accuracy for a-e letters and 0.6847 validation accuracy for a-y letters.

[1]School of Engineering and Computer Science, Baylor University, Texas, USA. Correspondence to: Abdullah Al Maruf <maruf_maruf1@baylor.edu>.

If we look into other sign languages, (Pigou et al., 2014) worked on Italian gestures. They considered a recognition system using the Microsoft Kinect, convolutional neural networks (CNNs), and GPU acceleration. The authors were able to recognize 20 Italian gestures with high accuracy. The predictive model was able to generalize on users and surroundings not occurring during training with a cross-validation accuracy of 91.7

3. Methodology

3.1. Data

We used the data set from the Kaggle platform (https://kaggle.com/grassknoted/asl-alphabet). Each alphabet has 3000 images (Figure-2), including 26 Alphabets as well as the Delete, Space, and Nothing gestures.

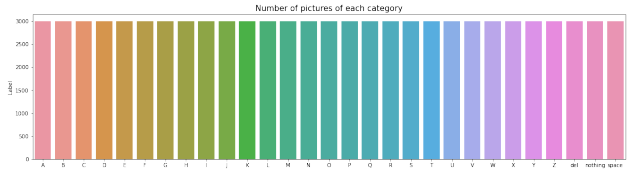

Figure 2. Number of images in each catagory in dataset

Despite the large number of images, the number of verities was very low. It would be preferable if we could find more data with different backgrounds and lighting, as well as different people.

Figure 3. Sample images and it's labels from dataset

3.2. Convolutional Neural Network (CNN)

According to (O'Shea & Nash, 2015), With the emergence of the Artificial Neural Network (ANN), the field of machine learning has taken a dramatic turn in recent years. One of the most impressive forms of Artificial Neural Network (ANN) architecture is Convolutional Neural Network (CNN). CNNs are mainly used to solve difficult image-driven pattern recognition tasks due to their detailed yet simple architecture.

The Convolutional Neural Network (CNN) is useful for extracting features from images. Due to the reduced number of parameters involved and the reusability of weights, CNN does a better fitting to the image dataset. In other words, the network can be trained to better understand the image's sophistication.

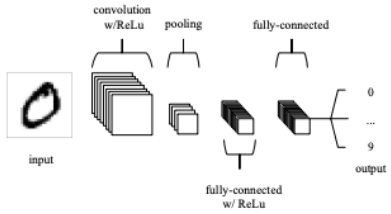

Figure 4. Basic architecture of CNN (O'Shea & Nash, 2015)

CNNs are made up of three layers: convolutional layers, pooling layers, and fully-connected layers, according to (O'Shea & Nash, 2015). A CNN architecture is formed when these layers are stacked. Figure-4 shows a simplified CNN architecture for MNIST classification. The output of neurons will be determined by the convolutional layer, and the rectified linear unit (ReLu) will apply a 'elementwise' activation function such as sigmoid to the output of the previous layer's activation. The pooling layer will then simply perform downsampling along the spatial dimensionality of the given input, further reducing the number of parameters within that activation.

3.3. Transfer Learning

Transfer learning is a machine learning technique where a model trained on one task is re-purposed on a second related task. Usually, these pre-trained models are trained against a large dataset and contain more layers. We used MobileNetV2 pretrained base model, which was developed at google, pre-trained on the ImageNet dataset with millions of images and 1000 classes of web images(Sandler et al., 2018).

4. Experiments

4.1. Training Model

We added two dense layer on top of MobileNetV2 model. One of these two is output layer. We divided the dataset into three parts. At first, we took 90% data for training and 10%

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Figure 5. CNN Layers in MobieleNetV2 model (Sandler et al., 2018)

data for testing. Further, we divided training data into train data set and validation datasets. We took 10% data of train data for the validation dataset.

We ran 5 epochs with 'earlystopping' callbacks such that if the validation loss is not improving for two consecutive epoch, the training process will stop while restoring the best weights so far.

While compiling the model we used 'adam' for optimization. Adaptive Moment Estimation (Adam) is a method that computes adaptive learning rates for each parameter. Also, for loss calculation, we used 'categorical_crossentropy'.

4.2. Real-time application

We developed an user interface that can detect hand gesture using the UI.

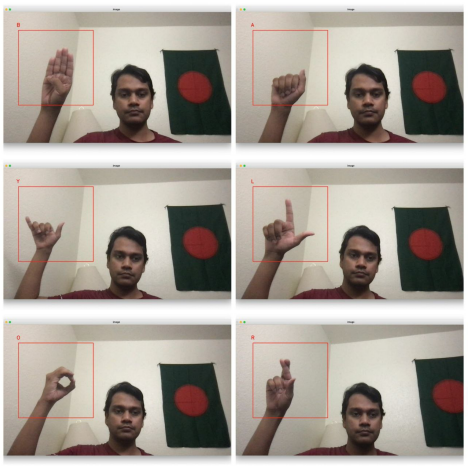

Figure 6. User Interface of real-time application

Figure-6 shows 6 different snaps of the application where it was able to identify the letters 'BAYLOR'. The application takes the portion of rectangle to run prediction and the most likely letter is shown on the left upper corner of the rectangle.

5. Analysis

5.1. Result

Figure-7 shows the validation and training accuracy and loss.

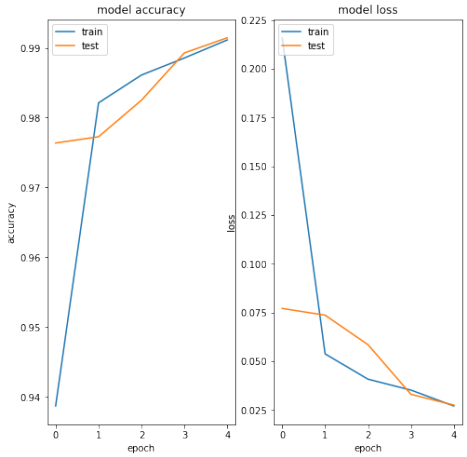

Figure 7. Training vs Validation accuracy and loss graph

We achieved 0.991 validation accuracy and 0.027 validation loss. We ran evaluation on the test set that the model never seen and it got 0.994 accuracy and 0.0227 loss. Table-1 shows the validation and training accuracy and loss for each epochs.

Table 1. Training and validation accuracy and loss

EPOCHS	TRAINING LOSS	VALIDATION LOSS	TRAINING ACCURACY	VALIDATION ACCURACY
1	0.219	0.082	0.935	0.973
2	0.055	0.043	0.981	0.984
3	0.039	0.054	0.986	0.981
4	0.035	0.053	0.988	0.983
5	0.029	0.050	0.990	0.984

After plotting the test result into a confusion graph, we got Figure-8.

Figure-9 shows some sample output of test set.

5.2. Real-time testing

While using the real-time application, the model was able to classify some letters very easily. But, for few letters which

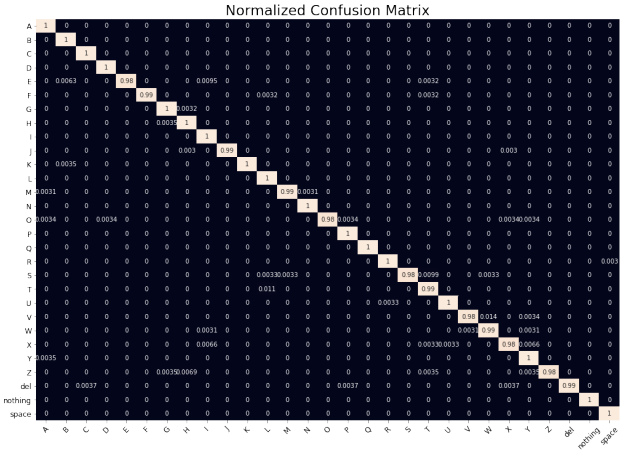

Figure 8. Confusion matrix on predictions of test set

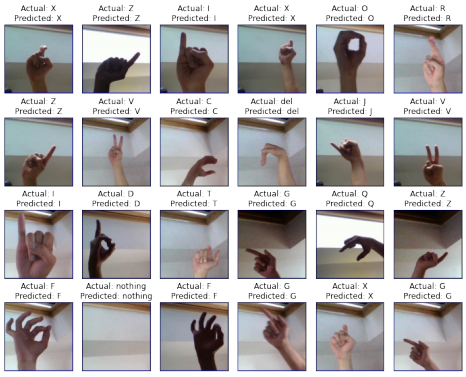

Figure 9. Sample predictions on test set

seems almost similar, it was having problem to identify correct word. One reason can be, the low varieties of dataset. Also the number of person was not more than two in the dataset which may cause the learning faulty.

6. Conclusion

This work shows that CNN with transfer learning is very powerful tool. One can utilize powerful pre-trained model to predict more accurate results in machine learning. The real-time application is a python script. But, the saved model can be used with almost any devices. As the new technology emerges, hopefully the communication gap between deaf and the other people will go away in upcoming days.

7. Future Work

As we have seen from the real-time application, we need more variation in dataset to tackle similar gestures. As tensorflow has capability to run model predictions on mobile, a mobile real-time application also can be implemented which will be more practical to use.

References

Garcia, B. and Viesca, S. A. Real-time american sign language recognition with convolutional neural networks. *Convolutional Neural Networks for Visual Recognition*, 2:225–232, 2016.

Mitchell, R. E., Young, T. A., Bachelda, B., and Karchmer, M. A. How many people use asl in the united states? why estimates need updating. *Sign Language Studies*, 6(3): 306–335, 2006.

O'Shea, K. and Nash, R. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

Pigou, L., Dieleman, S., Kindermans, P.-J., and Schrauwen, B. Sign language recognition using convolutional neural networks. In *European Conference on Computer Vision*, pp. 572–578. Springer, 2014.

Rivas, P., Rivas, E., Velarde, O., and Gonzalez, S. Deep sparse autoencoders for american sign language recognition using depth images. In *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, pp. 438–444. The Steering Committee of The World Congress in Computer Science, Computer . . . , 2019.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Zafrulla, Z., Brashear, H., Starner, T., Hamilton, H., and Presti, P. American sign language recognition with the kinect. In *Proceedings of the 13th international conference on multimodal interfaces*, pp. 279–286, 2011.