Interpreting Sign Language using CNN Models

Chun-Che Yang
National YangMing ChiaoTung University
Hsinchu, Taiwan
angusyang.cs09@nycu.edu.tw

Hsiao-Ting Shao
National YangMing ChiaoTung University
Hsinchu, Taiwan
ting0602.cs09@nycu.edu.tw

Abstract

Our purpose is to interpret sign language by detecting hand gesture instantly, and the machine will say out the sentence it recognized. The way to make the sentence is using the method of fingerspelling and some common gestures. We implement it by training the CNN Model with dataset of blurred gestures to predict the user's gesture. For more details and resource, please refer to the Github Repository

1. Introduction

In the recent years, there are more than 466 million people worldwide are hearing impaired, accounting for 15% of the world's total population. Additionally, the number of people with hearing impairment is expected to grow to 900 million by 2055. However, most people may not use sign language to communicate, and there are no interpretation systems launched yet for sign language users recently.

Sign languages can be quite different between each region, and most of words cannot be partitioned, which may lead to too large and diverse dataset, increasing the difficulty for AI to train model. In addition, sign languages don't merely consist of hand gestures, it also includes face expression, which also increase the difficulty for implementation.

For now, Google AI is working on a Real-Time Hand Tracking system implemented by MediaPipe, which is the base of recognizing gestures. The system uses machine learning to infer 21 3D keypoints of a hand from just a single frame. However, the dataset is hard to set up because it needs human to label those 21 3D keypoints in a frame.

Our goal is to try to overcome the problem of needing to set up labels for keypoints in a frame, we use blurred hand gestures instead to train CNN model [2, 3]. For recognition, we also blur the frame of gesture and use the model to predict what the gesture would be instantly, In this way, we decrease the complexity of setting up dataset.

2. Related Work

In this project, we use the CNN for supervised learning. The main difference between our project and Real-Time Hand Tracking system by Google AI is the processing of the dataset and detection. Real-Time Hand Tracking system use the MediaPipe to detect user's gesture, therefore, the images in dataset needs to be labeled the 21 3D keypoints by human, meaning it is hard to increase the diversity of dataset. In our project, we blur and convert images into black-and-white color by the functions in Open Source Computer Vision, decreasing the processing time and difficulty. By this simpler way, it can reduce the difficulty of increasing the diversity of the dataset. We think this way may be good for the development of sign language interpretation system since sign language has a rich vocabulary.

3. Methodology

3.1. Dataset

The gestures in dataset are frames which are blurred and then converted into black-and-white color through threshold function and converted into grayscale image [4]. In order to recognize gestures from left and right hand, we flip the frames as new data, so there are half original data and half flipped data.

3.2. Labeling and Classifying dataset

To train CNN model, we have to label our dataset. First, we label all the images to the represented alphabets or words. Second, We shuffle all labeled images. Finally, we take first 10/12 of dataset as train images, the next 1/12 as validation images, and the last 1/12 as test images.

3.3. CNN Model

Our CNN model can be separated into several main layers: 3 convolution layers, 3 pooling layers, 4 dropout layers, 1 flatten layer, 1 fully connected layer, and 1 output layer. The reason for adding dropout layer is to avoid overfit during training. We compile our CNN model with Adam op-

timizer as optimizer and categorical cross entropy as loss function.

3.4. Training

The input of the CNN model is labeled train images and use labeled validation images as validation data, with batch size is 500. The training process is 20 epochs, after training, evaluate the model with test dataset.

3.5. Prediction

Similar to dataset images, we blurred the gesture captured from camera and convert into black-and-white color immediately. Then, we take the blurred gestures as input for CNN model to predict. If the model predict the same word or alphabets over 10 times with a probability high than 80%, it will be added to the text which is showed on the screen.

3.6. Baseline

In this object, our baseline is to train the CNN model with original images without blurring and making it into black-white images. These images are stored in csv files, given in the form of labels and pixel value ranging from pixel 1 to pixel 784 which are 28 * 28 images. After training and testing, the error rate is around 10%. Then, we compare the error rate with the CNN model we trained. In order to compare both models, we make the number of dataset nearly equal, and the type of labels are only A to Z without J and Z. With training 20 epochs, the result shows that with quite the same training time, our CNN model has an error rate of 0.11%, where the baseline CNN model has a higher error rate of around 10%. Therefore, we think blurred images may be a better data for training and recognizing.

4. Experiments

4.1. Manipulation

First, user need to operate under a clean background and sufficient lighting to achieve better recognition results.

Then, user needs to cover the green area on the screen with hand for the program to recognized the hand. The camera would try to get histogram of your hand by recognizing the color and brightness of your hand comparing to the environment. In order to remove noise, we use Gaussian Filtering and then use Median Filtering. [5]

Finally, using sign language within the range of green box in the screen. During this process, the program calculate the threshold value for every frame and make the blurred and black-and-white color image to show and give CNN model to predict the gesture, then it will present the predicted result in the screen and the sentence can be pronounced by using the functions in the text-to-speech conversion library. [1, 6]

4.2. Discussion

During experiment, we found out noise in environment, such as brightness and background color, influences a lot for recognition. If the environment is too bright, the shadow part of hand may not be recognized. If background color is similar to skin color, it may recognize objects other than hand as part of hand gesture. For training model, if the number of filter is too many in a layer, it may lead to overfit of the model, reducing the accuracy of recognition.

4.3. Conclusion

Sign language recognition is a hard problem if we consider all the possible combinations of gestures that a system of this kind needs to understand and translate. Therefore, we want to divide it into simpler problems. The system Google AI is developing through MediaPipe may get a higher accuracy, however, it requires higher complexity for setting dataset. Hence, we use only camera, CNN, and blurred images to decrease difficulty of setting up dataset and recognizing hand gestures. Through this simpler implementation, we could also build a sign language interpretation system.

References

[1] K A Bhaskaran, A G Nair, K D Ram, K Ananthanarayanan, and H R NVardhan. Smart gloves for hand gesture recognition: Sign language to speech conversion system. In *2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA)*, 2016. 2

[2] Harsha Vardhan Guda, Srivenkat Guntur, Gowri Pratyusha M, Kunal Gupta, Priyanka Volam, and Sudeep P V. Hardware implementation of sign language to text converter using deep neural networks. In *Proceedings of the International Conference on Advances in Electronics, Electrical and Computational Intelligence (ICAEEC) 2019*, 2020. 1

[3] Uday Khati, Prajitesh Singh, and Achyut Shankar. Text generation through hand gesture recognition. In *Proceedings of the International Conference on Innovative Computing and Communications (ICICC) 2020, Available at SSRN*, 2020. 1

[4] S. Kolkur, D. Kalbande, P. Shimpi, C. Bapat, and J. Jatakia. Human skin detection using rgb, hsv and ycbcr color models. In *Proceedings of the International Conference on Communication and Signal Processing 2016 (ICCASP 2016)*, 2016. 1

[5] M. I. N. P. Munasinghe. Dynamic hand gesture recognition using computer vision and neural networks. In *International Conference for Convergence of Technology (I2CT)*, 2018. 2

[6] N Sriram and M Nithiyanandham. A hand gesture recognition based communication system for silent speakers. In *Proceedings of the 2013 International Conference on Human Computer Interactions (ICHCI)*, 2013. 2