ECE 228 GROUP 9 FINAL REPORT
SIGN LANGUAGE GESTURE RECOGNITION WITH DIFFERENT LIGHT SOURCE

Howard Chi, Yu Cheng, and Zhaowei Yu

University of California San Diego, La Jolla, CA 92093-0238,

ABSTRACT

The American Sign Language is a complete, natural language that helps those who suffer from deaf and hard of hearing. However, it's hard for people who are not familiar with ASL to communicate or make friends with the ASL users ,and also ASL is difficult to translate directly. Therefore, our solution to the problem is by using Image recognition with convolutional neural networks (CNN) to recognize the hand gesture and provide translation. Moreover, we expect that anyone could talk to any ASL users in any circumstances. Therefore, we adjust the intensity of validation data to perform an inadequate lighting environment. The result shows that without the histogram equalized of the data, it has low accuracy. However, if we histogram all the data first (including training data), then we still can get good results under dark circumstances.

1. INTRODUCTION

There are 500 thousand ASL users in the US Canada. Thus, ASL is a widely used language. However, it's hard for people who are not familiar with ASL to communicate or make friends with the ASL users. It's hard to translate directly using traditional methods such as text based translation since it does not contain any text, and learning this sign language could be time consuming. The solution we have to this problem is we can have a translator by using Image recognition with CNN to recognize the hand gesture and provide translation. Moreover, it's important to make the translator be capable of functions within any environmental condition, such as during the candle dinner or in the park at night. Therefore, we will adjust the intensity of the image to appropriate lighting conditions, and use histogram equalization for image processing to simulate dark conditions. The input of our system will be an image or a series of images. We then use CNN to output a predicted image. We expect that someone unfamiliar with ASL could understand what ASL users are saying using the translator easily and conveniently.

2. RELATED WORK

There are many related works for sign language gesture recognition by using machine learning methods, such as KNN, SVM or neural networks. We will only focus on neural networks here and why we choose CNN as the training model. [4] use CNN and RNN models to be trained independently. CNN is used for recognizing spatial features and RNN for temporal features. It shows that CNN has accuracy more than 90%. However, RNN only achieved 55%. [5] proposed his CNN architecture and compared it to the prior ANN model. In Particular, it was trained and tested with one dataset, and also tested with another dataset as a comparison (by another person). By his works, for the same dataset, ANN only has 84% accuracy, and the CNN model is 92%; for a different testing data, ANN is down to 77%, and the CNN is still at least 84%. Therefore, for the neural networks, we think CNN is a better option to fit into sign language gesture recognition. [1] and [3] shows the power of the CNN model. The former is training for ASL. It used 4 convolution2D layers and 3 maxpooling2D layers. The testing accuracy is 99.3%. [3] is trained based on Bengali Sign Language. The CNN model it proposed has 12 layers. It reached 99.86% test accuracy. As a result, we decided to use CNN as our training model.

3. DATASET AND FEATURES

ASL Alphabet Dataset from kaggle is used in this project. Each image is 200*200 pixels, and has 3 channels for RGB as our input data. The images are categorized into A to Z. 300 images are used for each category. 80% of images for training and 20% of images for validation data. Actually, there are many datasets containing ASL on the internet. We chose this one because it has the most kind of images. The dataset has 3000 images per letter, and each class has different size, angle, lighting or location of the hand gestures. We believe that it can give the most diversity of the input so that our model can fit most to the real world. Unfortunately, due to the limited memory on the datahub, we can only use 300 images per letter as our input. All the input images are first preprocessed so that the intensity levels are set to appropriate values. Then, we change the intensity levels of some images into their HSI form in order to imitate images with different lighting conditions.

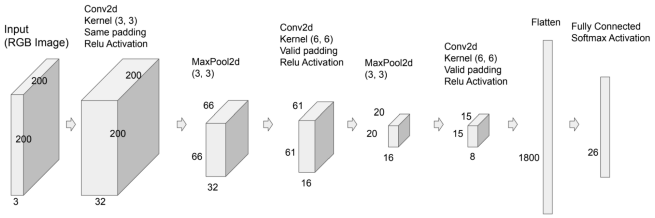

Fig. 1. Our CNN model

4. METHODS

Because of the nonlinearity introduced, we think that machine learning models have the potential to achieve higher accuracy. We use CNN as our training model. Unlike the conventional model such as least square regression model, CNN is a class of machine learning and deep neural networks. It has one or more convolutional layers that can help us do the classification of different gestures.

4.1. Model

Fig.1 is the convolution neural network model we used in this project. The inputs of the model are 200 by 200 RGB images and are gradually transformed into 15 by 15 by 8 units. 32 kernels are used in our first conv2d channel because we don't think so much information can be extracted from a 3 channel image that over 100 kernels must be used. Since we want to make the training process faster, only few data are used, and avoid overfitting due to a high complexity model, we didn't use a deep neural network in this project.

4.2. HSI

HSI respectively stands for hue, saturation and intensity. It's an alternative representation of the RGB color model. The hue component is the color itself in the form of an angle between 0 to 360 degrees. 0, 120 and 240 mean red, green and blue individually. The saturation describes how much the color is diluted with the white color. The range of the saturation is between 0 and 1. The intensity is about brightness, which is important in describing color sensation, 0 means black and 1 means white. It's also the key factor that we use to determine the lighting conditions.

4.3. Histogram Equalization

Histogram equalization is a technique to enhance contrast adjustment using the image's histogram. It is used to make the distribution of the pixels of the image on the intensity become closer to uniform distribution. We apply this method to RGB images by transforming it into HSI color space first to avoid dramatic changes in the image's color balance.

4.4. Method 1

The model is trained with data under appropriate light conditions. Before being fed into the model for classification, the test data are first histogram equalized so that the test data become under a better lighting condition. Since the lighting condition of the testing data gets better, the same model may be able to classify the data which were incorrectly recognized originally due to the bad lighting condition.

4.5. Method 2

No assumption is needed. Both the training data and the testing data are histogram equalized before being fed into the model. After the histogram equalization, all the data may become under similar lighting conditions. Therefore, a better performance may be achieved.

5. RESULTS AND DISCUSSION

5.1. Image pre-processing

Top left is the original image. Top right is the brightened image. If the intensity of the image is below 0.5, we adjust it by 0.1. In other words, the light intensity has increased 0.1. Bottom left is the darken image. We adjust the intensity by . For example, if the original intensity is 0.6, then it becomes 0.4. Bottom Right is the result of the histogram equalized original image. Top right and bottom left clearly shows that we can simulate real world lighting conditions through changing the intensity of the image.

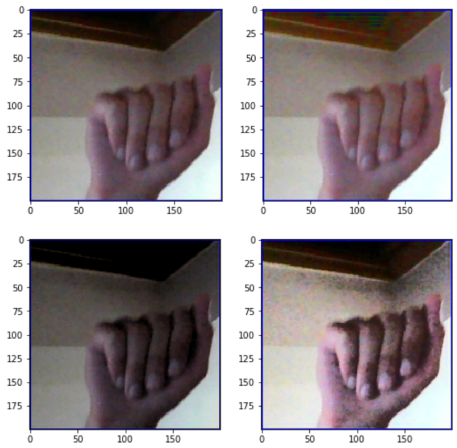

Fig. 2. Processed images

5.2. Image equalization

Right is the histogram equalization of brighten image. Left is the histogram equalization of darken image. We can see that there is not much difference between these two, even though their original images have obvious differences in the lighting.

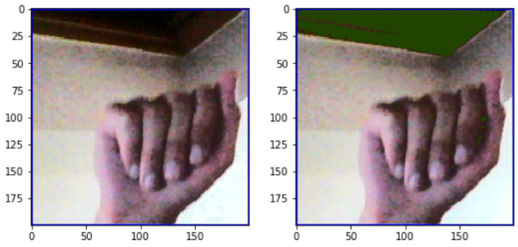

Fig. 3. equalized images

6. CNN TRAINING

6.1. Normal training and validation image

In Fig.4, we adjust the intensity levels of all the data . We can clearly see that both the training accuracy and validation accuracy is close to 1, and the loss is also low.

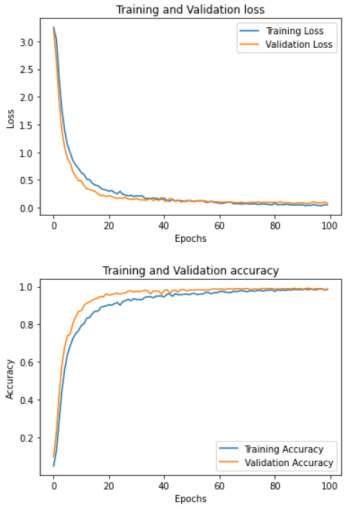

Fig. 4. Training and validation loss and accuracy of all data have appropriate intensity levels

6.2. Normal training and darkened validation image

In Fig.5, we decrease the intensity to imitate data under insufficient lighting conditions. The loss is high and the accuracy is extremely low. It tells us that we cannot just use CNN for image recognition when the training dataset doesn't contain data under various lighting conditions.

6.3. Normal training and histogram equalized darkened validation image

In Fig.6, we apply histogram equalization to the darken validation data. Due to the histogram equalization, the result is a little better than the previous case but accuracy is still low.

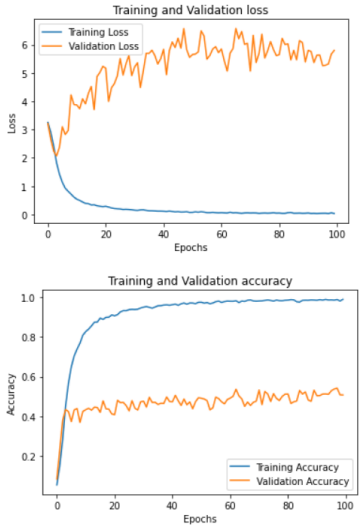

Fig. 5. Training and validation loss and accuracy of validation data has been darkened

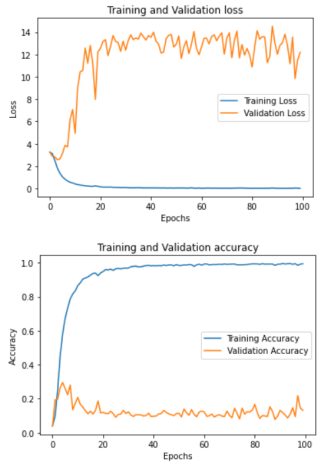

Fig. 6. Training and validation loss and accuracy of applying histogram equalization to the darken validation data

6.4. histogram equalized training and histogram equalized darkened validation image

Finally, we apply histogram equalization to all the data, including the training data and the darken validation data. The result clearly shows that after the histogram equalization, the validation is as high as the training accuracy. Also, the loss is down and close to zero. As the result, we can apply histogram equalization to all the input images first, then no matter where the users are, no matter how much lighting they are under, this CNN model can distinguish the letters under any environment and help us communicate with ASL users without any problem.

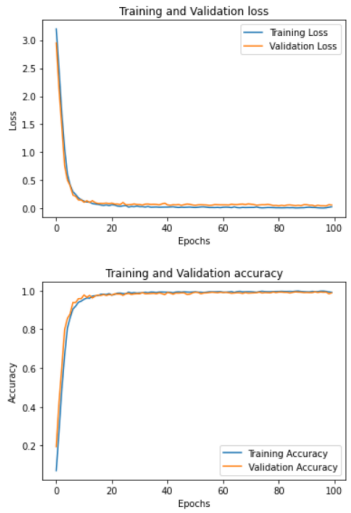

Fig. 7. Training and validation loss and accuracy of applying histogram equalization to the darken validation data training data

7. FEATURE EXTRACTION

A convoltional autoencoder has been trained on the training images and shown to be capable of reconstructing original images from the latent features.

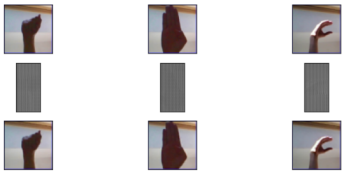

Fig. 8. Original, latent space and reconstructed images

Yet, as Fig.9 demostrates the accuracy training and validating the latent space representation of the bright data in the CNN model is low, potentially due to the CAE being not optimal. We would like to explore this issue in the future.

8. CONCLUSION

Histogram equalization makes the data under similar lighting conditions. When histogram equalization is applied on both training and testing data, they all become under similar lighting conditions. This results in great performance, and we can observe this improvement in the experiment.

9. CONTRIBUTION

Howard Chi: model training, write report Yu Cheng: histogram equalization, model training, write report Zhaowei Yu: model training, write report

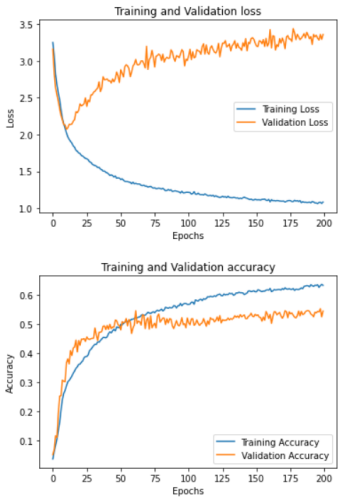

Fig. 9. training and validation accuracy and loss of the latent space representation of the image

References

[1]Goswami T., Javaji S.R. (2021) CNN Model for American Sign Language Recognition. In: Kumar A., Mozar S. (eds) ICCCE 2020. Lecture Notes in Electrical Engineering, vol 698. Springer, Singapore.

Garcia, Brandon, and Sigberto Alarcon Viesca. "Real-time American sign language recognition with convolutional neural networks." Convolutional Neural Networks for Visual Recognition 2 (2016): 225-232.

M. J. Hossein and M. Sabbir Ejaz, "Recognition of Bengali Sign Language using Novel Deep Convolutional Neural Network," 2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI), 2020, pp. 1-5, doi: 10.1109/STI50764.2020.9350418.

K. Bantupalli and Y. Xie, "American Sign Language Recognition using Deep Learning and Computer Vision," 2018 IEEE International Conference on Big Data (Big Data), 2018, pp. 4896-4899, doi: 10.1109/BigData.2018.8622141.

G. A. Rao, K. Syamala, P. V. V. Kishore and A. S. C. S. Sastry, "Deep convolutional neural networks for sign language recognition," 2018 Conference on Signal Processing And Communication Engineering Systems (SPACES), 2018, pp. 194-197, doi: 10.1109/SPACES.2018.8316344.

W. Okado, T. Goto, S. Hirano and M. Sakurai, "Fast and high-quality regional histogram equalization," 2013 IEEE 2nd Global Conference on Consumer Electronics (GCCE), 2013, pp. 445-446, doi: 10.1109/GCCE.2013.6664884.

10. CRITICAL REVIEW FROM GROUP 6

I thought this was a very cool topic to use machine learning for! I have never considered how different ASL is from a spoken/written language like English when used in ML. Was the primary goal of this research to auto-translate videos of ASL?
- Yes, since videos are just pictures in motion, we would feed individual frames into the model and recognize the letter.

I would like to hear more about the findings from your literature review. Why did the CNNs perform so much better than RNNs and ANNs?
- Due to the nature of CNN being a convolutional network, it performs better with image-like inputs and no flattening is required. Thus it performs better in this particular case.

I thought your explanation of why your topic is a good choice for ML could use a bit more explanation. Have there been any approaches to this problem that didn't work well? Is there a lot of data available for training?
- ML is an often-used method for image classification. And, in most cases, high accuracy is achieved. So, from other people's experience, we think that ML will be a good approach. Non-linearity is introduced by the activation function or by the kernel. Therefore, ML models do have the potential to achieve better accuracy when compared to linear methods, such as the least square classifier and the Wiener filter. Yes, there is much data available, but not a lot of data is used for training because of limited available memory.

For the second and third iterations of your results, could you create a training dataset that has both dark and bright images, then validate on another combined dataset? Or even combine "normal", bright, and dark images to create a training dataset with more variance? I think that would improve your results even before histogram equalization. Histogram equalization was a great tool to help clean your dataset. I think it would help with a variety of lighting conditions that may be present. Did the paper whose work you followed perform and sort of data cleaning? The example images all looked like they were fairly unobstructed and against a plain background.
- We would like to try that.

The model explanation was appropriate. It went in enough detail that I understood why you picked the CNN architecture, but you didn't waste time explaining aspects of the model that weren't critical. It also seemed like you assumed a baseline knowledge of ML (which I think was good).
- Thanks. :)

When you mention incorporating your autoencoder in future work, what do you mean? Would you change the input into your model to be the latent space representation of your original image?
- Yes we plan to do that. After training, we would feed the testing data into the auto encoder and use the latent space representation of the image for prediction.

I'm not sure how long the histogram equalization processing takes, it seems like that may be a limiting factor in quickly translating a lot of ASL images in various lighting conditions. Do you think there are any types of ML models that could eliminate that step? Or are better suited for this problem?
- Based on the literature survey we look at, it seems that CNN does the best job for ASL gesture recognition. But, for the histogram equalized image, there may exist other ML model suits better for this problem. We would love to try that.

Critical review from group 34 Why does the CNN model stand out from the others?
- See answer to problem 2

Please explain how to complete histogram equalization. Is it just making the sum of the intensity of each channel identical?
- $(L-1)\int_0^X f_x x\,dx$**, where Y is the pixel value after histogram equalization and X is the original pixel value. If an RGB image needs to be equalized, the image must first be transformed into its HSI form. Then, histogram equalize only the intensity channel or both the intensity and saturation channels. Finally, transform back to RGB form, and the result is the histogram equalized version of the original image.**

Why is the accuracy of the result lower than the referenced paper? Is there any way to further improve the result?
- First, we use different dataset, and the model we build is different, so it's reasonable that we get different results and accuracy. For the referenced paper, they actually did a great job, that's why we decided to use it as reference and learn from it. Secondly, definitely Yes. Even though currently we get good accuracy, the time for training could be too long, and we also can shrink the size of the image to improve the running time. We plan to use an auto encoder to do the feature extraction.

Have you tuned any parameters? How will parameters influence the result?
- We have tested different types of loss functions and the Kullback-Leibler divergence functions the best compared to other types of loss function, such as mse.

11. CRITICAL REVIEW FROM GROUP 35

While it is obvious why Machine Learning is important for the task, the group could go into more details about why ML is important because they just briefly mentioned it without explaining why.
- See response to problem 3.

Adding more bullet points about important information that the group talks about in every slide will make the viewer have an easier time following their storyline. For example, writing out how intensity levels were changed in the images in the HSI form in the preprocessing stage in the 'Dataset" slide will be very helpful.
- Fair enough

The group should spend more time talking about their model and the explanation was shallow.
- Thanks for the advice.

The presentation style/design was bland and could use better design to be closer to the project theme of American Sign Language to enhance the presentation.
- We prioritized the content of the presentation over flashy pages.

Other implementations that the group tried could be added with their details of their implementations and results. For Example, any other network architectures used. Also, you could add the results of the same network with different hyperparameters and how you tuned the hyperparameters to get this performance.
- The main purpose of this project is to find out how we can use histogram equalization to make the accuracy higher when the training dataset doesn't contain data under different light conditions. The hyperparameters are not what we focus on. So, we didn't wouldn't add other results.