ASL Hand Gestures Classification

Mekhak S'hoyan
Department of Applied Statistics and Data Science
Yerevan State University

Abstruct: **A hand gesture classification system provides a natural, innovative and modern way of non-verbal communication. It has a wide area of application in human computer interaction and sign language. The intention of this work is to create a hand gesture classification model using machine learning (deep learning) techniques.**

1. INTRODUCTION

American Sign Language (ASL) [1] is one of the main forms of communication among the deaf communities in United States and Canada.. Our aim is to develop a translator that can recognize hand symbols from a segmented hand gesture image and find the corresponding meaning. This would be useful aid for communication between deaf or mute people and those unfamiliar with sign language.

The ASL alphabet contains all 0-9 numbers and A-Z letters (total 36 different hand gestures). Some numbers and letters look familiar enough. Despite of that we tried to train a machine learning model which perform reasonably good on those familiar cases also.

2. DATASET AND FEATURES

We have considered the dataset from Masset University [2]. The dataset contains 2520 RGB images - 70 images for each hand gesture. The images are segmented and cropped (figure 1). The ASL hand gestures for letters can be seen in figure 2.

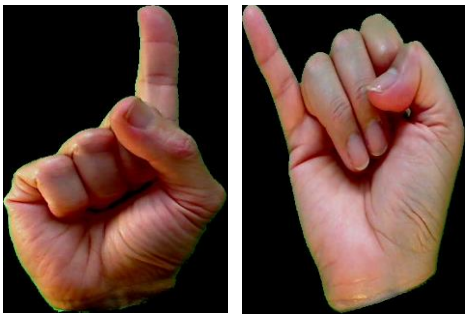


Figure 1. Examples of dataset images.

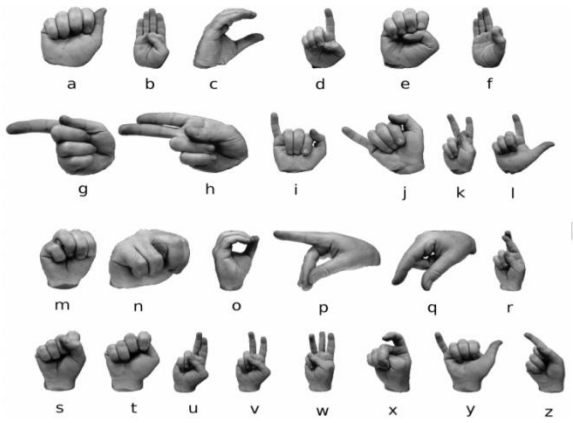


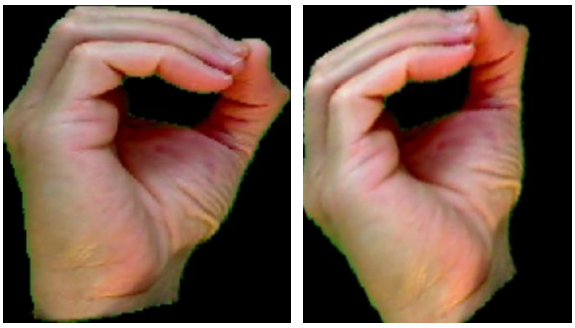
Figure 2. ASL hand gesture for a-z letters.

The dataset contains images of 5 individuals with variation of lighting conditions and hand postures.

The images were taken at a certain angle of rotation. So the dataset can be extended by tilting the original images. In practice, the person making the gesture may tilt the hand slightly, which affects the position of the

hands and fingers. So the tilted images generated by data augmentation process would be completely reasonable and equivalent to the originally tilted images.

We have considered augmenting the dataset by rotating the original images by 40 degree maximum rotation range. The example of an original image and its rotated versions are presented in figure 3.



(a) (b)

Figure 3. (a) Original image; (b) Augmented one from (a) using 40 degree maximum rotation range.

Images in the dataset had different sizes (both width and height were from the range 140 – 180) so the images are resized to have dimensions 150x150. The pixel values were scaled to the range [0, 1] by dividing the values by 255 (this was not a necessary step for random forest model, but it is required and useful for others: see below).

No additional data preprocessing (image cleaning) was done since the images are well segmented, clear enough and non-noisy. The images are taken under good lighting conditions.

The dataset was splitted into train, validation and test sets with number of images for each class 46, 10, and 14 correspondingly. Thus total number of images in train, validation and test sets are 1656, 360 and 504 correspondingly.

3. EXPERIMENTS AND RESULTS

As the "No Free Lunch" theorem states, there is no one model that works best for every problem. The assumptions of a great model for one problem may not hold for another problem, so it is common in machine learning to try multiple models and find one that works best for a particular problem. Our experiments showed that this was true for the case of our problem too.

It is well known that convolutional neural networks take up a domain position in computer vision classification problems. But it turns out that the simple logistic regression model performs equally good compared to the fine tunes VGG16 convolutional neural network in the case of our problem (at least based on our experiments). Let's discuss the details of the experiments we have made and the experiments results for each of the algorithms we tried for this problem.

We have tried 4 different algorithms for the problem:

- Logistic regression model,
- Random Forest model,
- A simple Convolutional Neural Network with custom architecture,
- Fine-tuned VGG16 Convolutional Neural Network.

Since the data is well balanced the accuracy score has been chosen as a model evaluation metrics.

Since the Logistic regression and Random Forest are relatively tiny models than CNNs and the training and validation process for them took reasonably short time, a grid search cross validation with 3 folds has been performed for the hyper parameter tuning and model evaluation for them. For CNN models simple hold out validation was performed using validation set described in previous section.

First, the logistic regression model has been tried as a simple linear model. A cross validation grid search has been performed for model evaluation and hyper parameter tuning (penalty, regularization strength). The cross validation accuracy was almost 98.2%.

Then, the random forest algorithm has been fitted with Gini impurity splitting criteria. A cross validation grid search has been performed for model evaluation and hyper parameter tuning (min samples leaf, max depth). The cross validation accuracy was almost 95.5%. Since the random forest supports the impurity based importance estimation of the image pixels let's see which parts of image pixels make high impact on prediction (fig. 4).

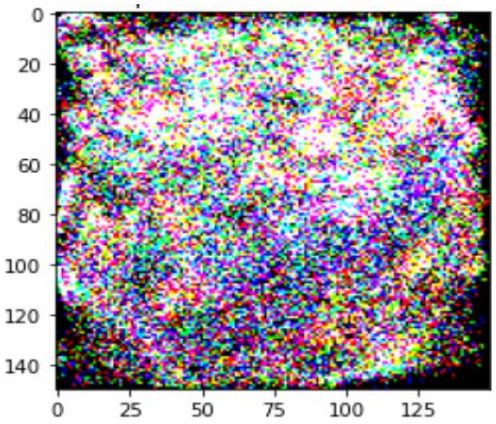
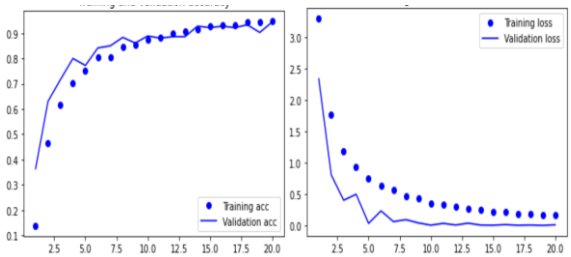


Figure 4. Pixels importances with random forest.

We can see that the random forest algorithm threats the pixels as unimportant features situated near the corners of the image. This is usual since the images are cropped and the hand gesture part fits in the image close to the borders.

In general, the random forest and logistic regression algorithms lack of flexibility for feature processing and feature extraction unlike the convolutional neural nets. In the image classification problem generally we need to do feature extraction somehow. We can achieve this before training the model using some image procession techniques like image sharpening, edge detection, etc. But this is like a guessing what kind of filter to choose to get good features for machine learning algorithms. The best way to automate the feature extraction process is to use convolutional neural nets.

A simple convolutional neural net has been used with 4 convolution + max pooling layers with 3x3 convolution filters, 2x2 max pooling filters and 2 fully connected layers with relu activation function. The 4 convolution layers have the depths 32, 64, 128, 128 correspondingly. The first fully connected layer has 512 nodes with relu activation. Finally, the previous layer has 36 nodes with softmax activation function. The training has been stopped after 20th epochs since the validation loss started to increase. The test and validation losses and accuracies during the training are given in figure 5.



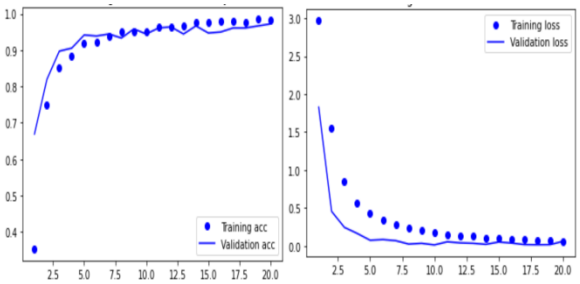
(a) (b)

Figure 5. (a) training and validation accuracy vs epochs. (b) training and validation logloss vs epochs.

This model showed almost 95% validation accuracy. Most probably the reason of not a high accuracy of this model compared to the logistic regression and random forest models was that the training data was not big enough to train convolutional neural network from scratch even though we augmented the data. Maybe we were needed to use other augmentation methods like rescaling and cropping, shearing, etc. This gave us insight to use transfer learning method.

The VGG16 model was chosen as the convolution base part. Then a fully connected

layer was added on top of the convolution base part with 256 nodes with relu activation. Finally, the previous layer has 36 nodes with softmax activation function. We froze the convolution base part so it became just a feature generator on top of which we trained the last 2 layers. The test and validation loss and accuracy are given in figure 6.



(a) (b)

Figure 6. (a) training and validation accuracy vs epochs. (b) training and validation logloss vs epochs.

This model showed 97.6% validation accuracy. It still has some improvement rooms (adding some regularization, more data augmentation, adding more layers on top of VGG, etc).

Finally, all 4 models were tested on test set. The validation and test accuracies of all 4 models are shown in table 1.

Model	Logistic Regression	Random Forest	CNN	VGG16 transfer learning
Validation accuracy (%)	98.2	95.5	94.444	97.2
Test accuracy (%)	98	95.6	94.40	97.6

Table 1. The validation and test accuracies of all 4 models.

All the codes and experiment results are placed in the github repository [3].

4. CONCLUSIONS

So, the results of experiments show that the best performing model on our dataset is the logistic regression model. One conclusion that we made was that with small datasets a simple linear regression model may be good than convolutional neural nets. The reason of this is that in small datasets the images (classes) might be more or less linearly separable. But when we have big dataset and nonlinearities the CNNs will outperform and achieve a better fit.

Since our dataset was not big enough the transfer learning method was useful. In fact, out dataset was not enough to train convolutional neural net from scratch and get high accuracies.

Regarding to the random forest model, we think that one reason that its accuracy was not high enough is that in the case of our images the informative features differ from one image to another (the informative features / pixels in general must be the edges of the hand). So it is hard to choose some fixed features (pixels) that will be equally informative for images of different classes.

REFERENCES

[1]https://www.nidcd.nih.gov/health/american-sign-language
[2]https://www.massey.ac.nz/~albarcza/gesture_dataset2012.html
[3]https://github.com/Mekhak/asl_handgesture_classification