Hand Gesture Recognition using Deep Learning

1 Introduction

Sign language is an essential part of communication used by deaf or mute people. Task is to design a classifier to predict hand gesture as an English alphabet letter (except J and Z, since they require motion) as used by American Sign Language (ASL) (Figure 1a). This can be potentially useful for improving the communication with deaf or mute people. Objective of this project is to design a model that can identify hand gestures and classify into corresponding English alphabets. We have training data consists of 27455 cases and validation data consists of 7172 cases. Each of this case is an images of 28×28 pixels. Figure 1b shows different letters expressed by hand gestures in 28×28 pixels. This dataset is available on Kaggle.

2 Data Creation

This dataset consists of hand gestures from multiple users with different backgrounds. Further this dataset was processed through a pipeline of different brightness, resizing, rotation and various filtering (e.g. 'Mitchell', 'Robidoux', 'Catrom', 'Spline', 'Hermite'). Because of the tiny size of the images, these modifications effectively alter the resolution and class separation in interesting, controllable ways.

3 Data Wrangling and EDA

This step consists of inspection of collected dataset and to identify any potential problems (e.g. missing attributes, outliers, duplicate entries etc.). In this dataset, their are total 786 pixels available for each image. Both training and validation data set do not have any missing entries or any duplicate entries. And After this we move to exploratory data analysis. Each image represents a 28×28 size image. To do sanity check I also plotted few of the images (Figure 2). Response variable in the training and validation data are numbers from 0 to 25 indicating English alphabets. It is a multi-class classification problem where 0 corresponds to alphabet A, 1 corresponds to alphabet B and so on.

Next, I create a count plot (Figure 3) of the training dataset to find out the distribution of different classes. From the count plot we observe that almost all the classes are uniformly distributed within a range of +/- 100.

4 Data Preprocessing

In this step I prepare dataset for modeling purpose. I first separate the response variable from training and test data. Next, I reshape each image into 28×28 matrix. Next, I use LabelBinarizer package from sklearn package to binarize the labels/response variable. I now split the training data for training and testing. I use 70% of data for training purpose and 30% for testing purpose.

(a)

(b)

Figure 1: American Sign Language Letters **(a)** in high resolution and **(b)** in 28×28 pixels used for training

5 Modeling with CNN

I used 3 layered CNN for model building. In almost 15 epochs I got accuracy of 100% on training and test data. In Figure 4, I show the accuracy and loss plots for 50 epochs.

Next I validate this model against the validation data. I get an accuracy of 0.86 with the trained model.

6 Conclusions and Future Work

By using a CNN based model, I am able to get high accuracy on the validation data. Next step would be to use more dataset for training to further improve the performance. Also, another future work would be to integrate hand gesture recognition model into an app.

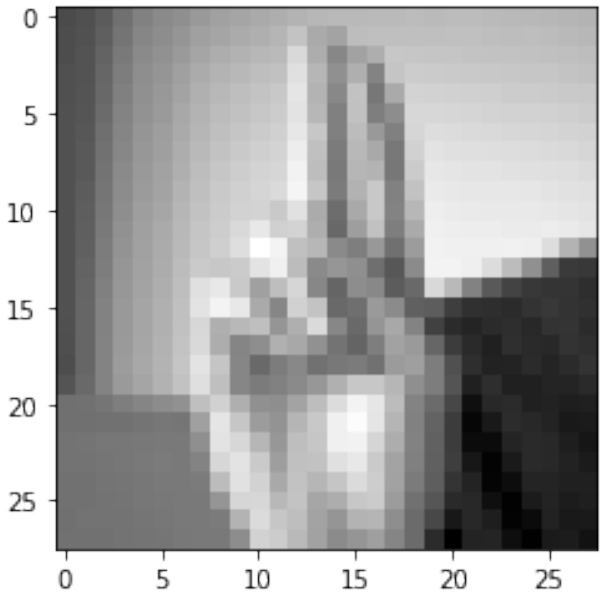


Figure 2: An image from training data

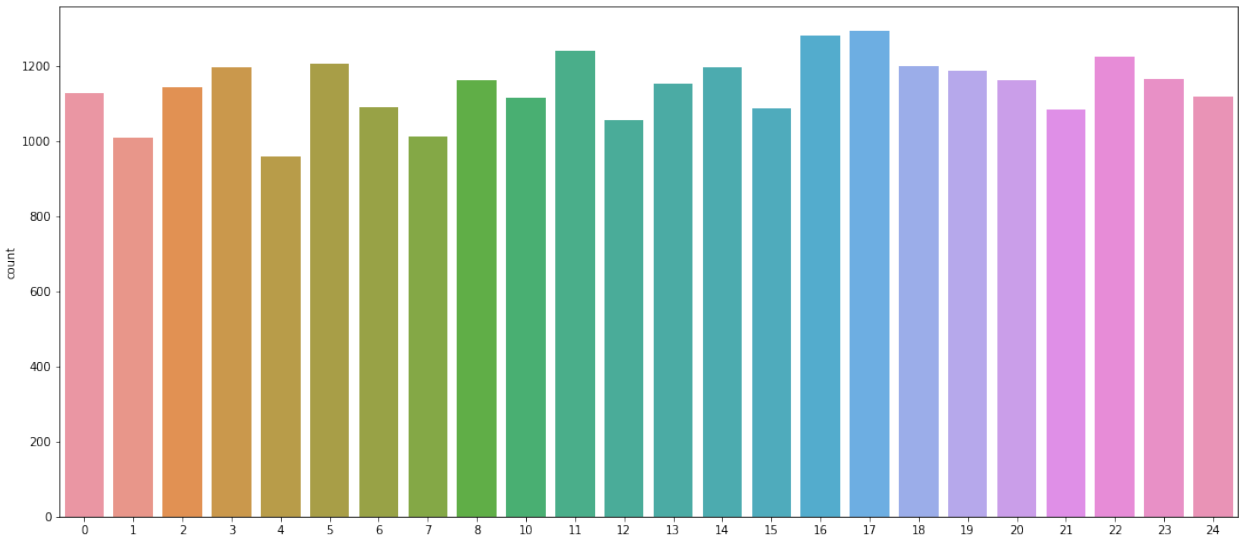
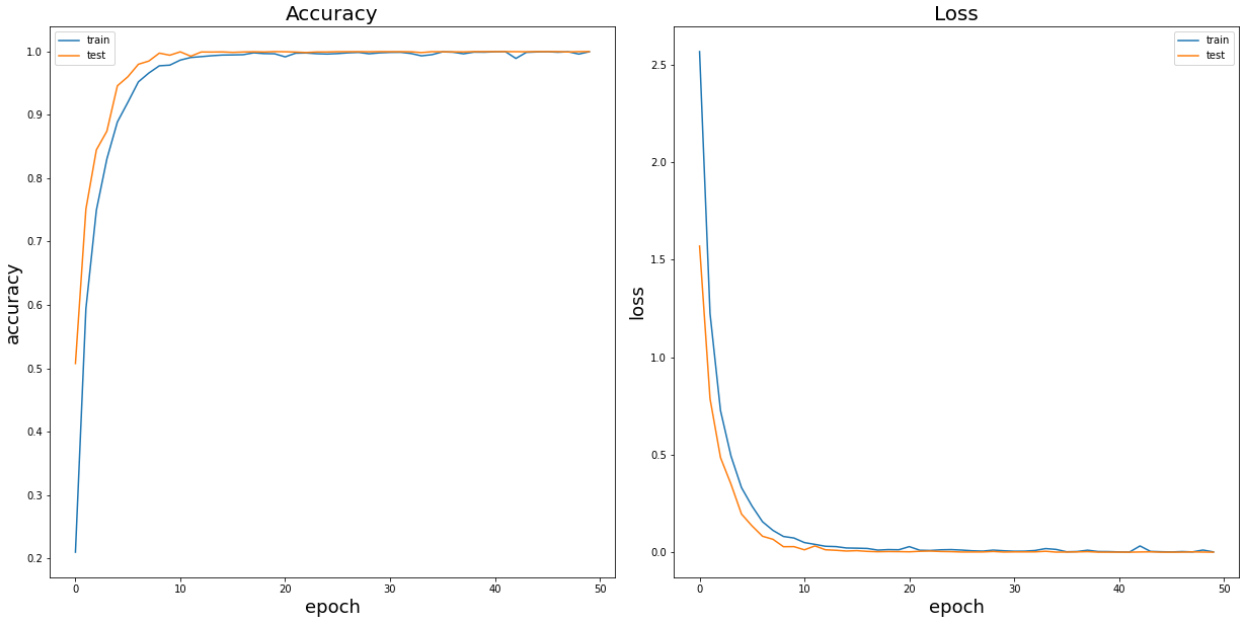


Figure 3: Count plot of different classes



(a) (b)

Figure 4: **(a)** Accuracy and **(b)** Loss variation with epochs