ELECTRICAL & COMPUTER ENGINEERING
TEXAS A&M UNIVERSITY

ECEN-765 Final Project Proposal

Drawing Recognition based on Google AI online game *Quick, Draw!*

Shaowen Zhu
UIN: 526005022

Date: Sept. 12th, 2018

I. Introduction and Motivation

Not long ago, there was a cellphone game named *DrawingGuess Go!* prevailing in China through the Chinese popular communication APP WeChat. This game was the first WeChat software developed by Google AI and based on a previous Google online game named *Quick, Draw!* . The game asks players to rapidly (in 20s) scrawl a picture of an object and then guess what the drawings represent through a neural network artificial intelligence.[1][2] While Google has constructed a model of NN (neural network), every time the plays pursue the game, it will also provide more training data to increase the model, which can be regarded as an online learning process. In this way, Google has also built the world's largest doodling data set with 50 million drawings across 345 categories.[3] Fortunately, this huge data set was posted on GitHub and well classified, so others can also build a classification model with 345 labels on it. In this project, my goal is to develop a similar system that can recognize the objects in doodling. I would like to construct my own classification model through NN or CNN (convolutional neural network). As the data set can be so huge that it is out of the computing capacity of our PC, I may just choose a part of the set with less labels. In this process, I will learn to represent the pictures through numerical forms and realize the mechanism of NN or CNN. By separating the data set into training set and testing set, I can evaluate the performance of my model, and therefore I can also try some different algorithms to make comparison.

II. Methodology and Resource

As mentioned before, Google has posted their data set on GitHub and we can get it from the repository of HalfdanJ. The link is appended in the references.

For the algorithm, I will learn the mechanism of NN and CNN first and figure out some critical factors, such as the number of layers I would like to choose. As scikit-learn[4] and TensorFlow[5] also provide many modules of such algorithms, it can be more convenient to construct my model, and I can also try different algorithms. To evaluate the performance, I would like to try two kinds of error estimation methods, directly separating the data set into training set and testing set, and cross validation method.

III. Expectation

The basic expectation for this project is to construct a complete classification model based on the data set I got, and the classifier can be used to recognize the objects in others' drawings. Besides, I will also evaluate the error of my classifier through the two methods mentioned above. By using different algorithms and setting different parameters, I will compare their performances in different conditions.

If the basic expectation is achieved, I would like to build an interface that players can interact with my classifier. Based on their new drawings, I can improve my model through online learning process. What's more, as I have got a huge number of doodling data of different categories, I can mix the data together and then sort them through some unsupervised learning algorithms, on which I can compare the performance of these algorithms in such state.

IV. Process Plan

1. Now ~ Sept. 30th: Learn the algorithms I will use and choose an appropriate amount of data as my training set. Figure out the outline of my model structure.

2. Oct. 1st ~ Oct. 31st: Construct the classifier and realize its fundamental function. Evaluate its error. Finish the midterm report.

3. Nov. 1st ~ Last week of classes: Try different algorithms or set different parameters, and make comparison. Try to reach the further expectation. Prepare for the presentation.

4. After Presentation: Finish the final report.

References

[1] Wikipedia, *Quick, Draw!* https://en.wikipedia.org/wiki/Quick,_Draw!#cite_ref-Wired_2-0.

[2] Burgess, Matt (16 November 2016). "You can now play a Pictionary-style game called Quick Draw against Google's AI".

[3] HalfdanJ, GitHub, https://github.com/googlecreativelab/quickdraw-dataset#the-raw-moderated-dataset.

[4] http://scikit-learn.org/stable/

[5] https://www.tensorflow.org/