

ANLY 590 Project Proposal

November 8, 2018

1 Team

This project will be teamed up with 3 people. Team members include Cheng Zhong, Ze Zheng, and Ye Zhang.

2 Project's Goal and Objectives

- The very first thing when we step into the deep learning area is to think about how can we use deep learning techniques to classify images. After we get through the MNIST example, we want to think more about image classification. Since three of us are definitely dog lovers, we are wondering is it possible for us to identify the breed of a dog through a simple image. Hence this project will mainly focus on image classification, especially, determine the breed of a dog in an image. The goal of this project will be to create a classifier capable of determining a dog's breed from a photo. The expected result of this project would be to deploy a model that will successfully identify the breed of a dog in an image within a reasonable loss range and accuracy.
- The motivation for doing this is due to some similarity between different breeds of dogs. For example, it is even very difficult for the human to distinguish Silky Terrier and Yorkshire Terrier. Thus it would be nice if an algorithm can identify this so we are very curious about that.

3 Data

The data we used for this project is originally from the Stanford Dog dataset and it was modified by Kaggle. The dataset includes a training set and a test set of the image of dogs. Also, the dataset includes a CSV file that contains all the labels for dogs and each image has a filename that is its unique id. The dataset comprises 120 breeds of dogs and including the following files:

- **Train.zip** - the training set, includes images for the breeds of dogs (more than 10000 images)
- **Test.zip** - the test set, includes images for breeds of dogs (more than 10000 images) and we will predict the probability of each breed for each image
- **Labels.csv** - the breed for the images in the train set (includes id column and breed column)
- Original Link: <http://vision.stanford.edu/aditya86/ImageNetDogs/>
- Kaggle Link: <https://www.kaggle.com/c/dog-breed-identification/data>

The possible limitations of the data might be there is no label for the test.zip to verify our result unless we submit our result to Kaggle competition in order to get a result. One way to solve this issues could be separate the training data randomly as two parts. One part treat as the new training data and the other part treat as the test data and then calculate the result. The reason we are choosing this data source is that the dataset is originally from Stanford University and it was carefully labeled. Also, as the data for Kaggle competition, it is reliable and good to use. Since we are doing a supervised problem which is actually a classification problem, we treat each image as a pixel matrix and transform it to put it into our model and the output would be a probability of each breed.

4 Assessment Metrics

- Since we are dealing with a multi-class classification problem here, the result will be evaluated on Multi-Class Log Loss between the predicted probability and the observed target. The reason chooses log-loss function here is because log-loss function heavily penalizes classifiers that are confident about and incorrect classification. For example, if the classifier assigns a very small probability to the correct class then the corresponding contribution to the log-loss will be very large indeed. Then it will be going to have a significant impact on the overall log-loss for the classifier. Hence the loss function and assessment metrics use here will be log-loss value. The smaller log-loss value will indicate a better prediction.
- Again, since this is a classification problem, the baseline dataset should be the test set used here.
- The base model used as the baseline is the pretrained VGG16 model in keras.
- As we all know that CNN has a very decent performance on image classification, we are also trying to build a CNN model to access the prediction.

5 Approach

- The approach of this project will mainly focus on the convolutional neural network (CNN). The reason we choose CNN is that CNN has a decent performance on image classification and CNN will learn to recognize components of an image (like lines and curves) and then learn to combine these components to recognize larger structures. The possible limitations of this approach might be the size of the training data is still not as big enough for a CNN. Also, the hyperparameter tuning is non-trivial and it might be cost effective.
- For this project, we will run our model locally. However, it might be faster to run it on the cloud but it might also require more set up procedure and cost.
- The API we use will be Tensorflow and Keras.