# Machine Learning Engineer Nanodegree

## Capstone Project

Daisuke Kuwabara
September 11th, 2020

## I. Definition

### Project Overview

This is the project “Dog Breed Classifier” using the Convolutional Neural Networks(CNN). In this project, the algorithm will identify the dog breed given an image of a dog. When the algorithm receive the image of human, it extrapolates its learned experiences and identify the resembling dog breed.

### Problem Statement

For this project, the main goal is classifying the dog and human images with dog breed respectively. To achieve this goal, several algorithms are implemented. The output is defined by each algorithm using conditional statements. 60% accuracy is set as the border line for the final output.

### Metrics

Since this task is simply the classification task, accuracy rate is used as metrics in this project.

# II. Analysis

## Data Exploration

Datasets include default images folder of dog an human images. “dogImages” folder contains 13233 total dog images. This folder are divided into three, train, valid, test folders. “lfw” folder contains 13233 total human images. Dog images and human images are used as inputs in this algorithm. “my_images” folder containing several images is used for testing the trained algorithm.

## Exploratory Visualization

There is no visualization except the images.

## Algorithms and Techniques

As I mentioned before, several algorithms are used in this project. First of all, OpenCV's implementation of Haar feature-based cascade classifiers is used to classify the human images. Secondly, Pre-trained model, VGG-16 by Pytorch is implemented to classify dog images. Finally the original algorithm is implemented using Transfer Learning.

Dropout used in the training process prevents the coadaptation of neurons.

Reference: https://arxiv.org/pdf/1207.0580.pdf

## Benchmark

I used the research paper, “Modified Deep Neural Networks for Dog Breeds Identification” by Shane Vahidnia, Stanford University as my benchmark.

http://cs231n.stanford.edu/reports/2015/pdfs/fcdh_FinalReport.pdf

# III. Methodology

## Data Preprocessing

For the data preprocessing purpose, Data Argumentation is the way to simulate the invariant representation. In the algorithm, resizing the image and rotation of the image. We expand the data and avoid the overfitting by generalizing the training data. We transform the Numpy arrays into the tensor by ”ToTensor” so that GPU will run on them. Normalization is used for the purpose of scaling.

## Implementation

Implementation is made in the following order.

First, Human detector is created importing the algorithm from OpenCV’s module. Then,Dog detector is created using VGG-16 model by Pytorch. This model will run on GPU. These model have high accuracy rate.

After that, I created CNN architecture to classify the dog breed.This algorithm performs quite poorly since it only contains thousands of images which is far from sufficient amount to train the algorithm. Plus, algorithm needs to be more complicated with more layers.

Transfer Learning is applied to create a new CNN by modifying the output layer. Note that the training process takes a time even with GPU.

## Refinement

I set the Epoch as 20 but around the end of training, while Training Loss is converging, Validation Loss remains same, or rather increased. I can try different number of Epoch such as 15 to avoid overfitting.

For the implementation, the condition is used to classify the images. This is still possible since the classification task is not complicated as a whole.

# IV. Results

## Model Evaluation and Validation

It has more than 80% of accuracy rate. This is really good accuracy even compared with other model introduced in the paper. However, special attention need to be paid to the number of inputs and outputs every time models are compared.

## Justification

It is over 60%, so it passed the benchmark

# V. Conclusion

## Free-Form Visualization

## Reflection

## Improvement

For Dog Image classification, adding more layers will potentially improve the initial accuracy of the model. Also, I can use the confusion matrix and so for the evaluation. I'm wondering things are much more complicated if there is cat classifier in this project.