# Machine Learning Engineer Nanodegree

# Capstone Proposal By: Felicia Tamasovics

# Convolution Neural Networks - Dog Breed Classification

## Domain Background

This project aims to identify dog breeds using Convolutional Neural Networks (CNN). Throughout this course, I have learned about Image Classification problems within Deep Learning Models. I have decided to explore this more in this capstone as CNN fascinates me as I use it on a daily basis to open my iPhone using facial recognition.

An individual may want to determine the breed of a dog for many reasons. One particular reason could be in determining common behavioural traits for that specific breed. [1] For example: The following breeds have similar physical traits: Beauceron, Rottweiler, Doberman, however differing temperaments. Beaucerons are commonly known for being loving, while Rottweilers have been known to be fearless and protective. [2] Since dogs have been breed for various reasons, it is evident that some are better as a service dog or a therapy dog. It could also be beneficial to know how much physical activity and attention the breed requires. Taking care of various dogs in my personal life, I have adapted to their needs: the Doberman requires lots of physical activity so he gets taken on lots of walks/runs, whereas the Maltese Chihuahua does not so we play fetch, or short walks for activity.

I also thought it would be fun to see what dog I most resemble!

## Problem Statement

This project will determine the breed of a dog using Machine Learning's Deep Learning CNN. The trained model will determine:

- an estimate of the dog's breed if a dog is detected
- an estimate of the dog breed that is most resembling if a human is detected
- state not a dog or human if neither a dog or human is detected

As with every model, it is important that it has a good performance. The notebook has specified that the CNN must attain at least 60% accuracy on the test set.

## Datasets and Inputs

There are two different datasets, human and dog images, both provided by Udacity.

Dog images:

- 8351 total images: 6680 train, 836 test, 835 validation
- 133 different breads
- Inconsistent number of photos for each breed
    - 3-10 photos of each breed in test and valid
    - 40-80 photos of each bread in train

Human Images:

- 13233 total images

Code for Graph Below:

```
dog_files_test = np.array(glob("/data/dog_images/test/*"))
numdogpic = []
dogbreed = []
for counts, spec_dog in enumerate(dog_files_test):
    spec_dog = spec_dog.split('.')[1]
    dogbreed.append(spec_dog)
    numdogpic.append(len(spec_dog))

import seaborn as sns
fig, ax = plt.subplots()
fig.set_size_inches(20,25)
ax = sns.barplot(x=numdogpic, y=dogbreed)
plt.ylabel("Dog Breeds")
plt.xlabel("Number of Photos")
plt.title("Distribution of Dog Breeds")
```

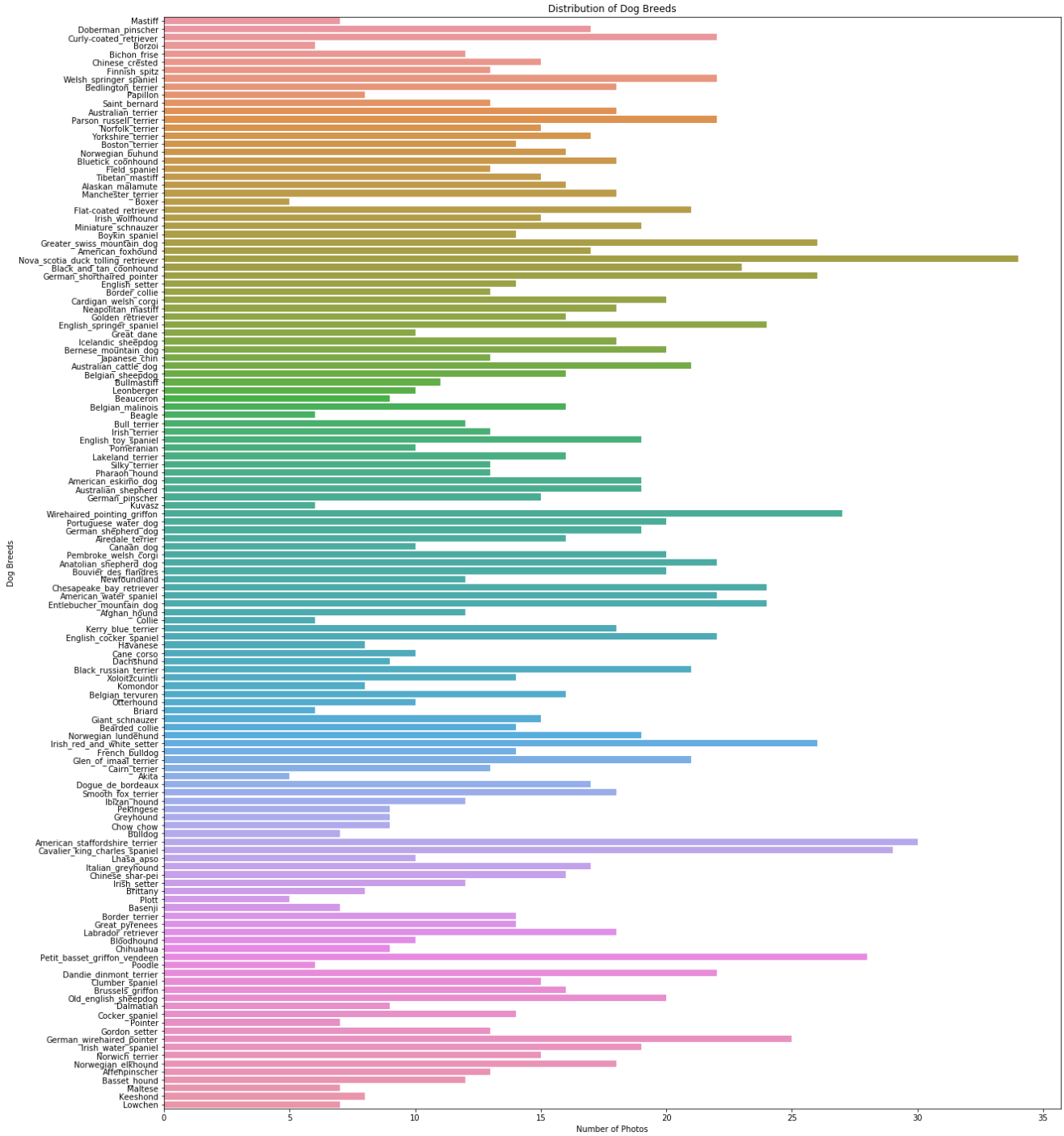


Figure 1: Distribution of Dog Photos for Each Breed for Test Dataset

Since the goal is to identify the breed of a dog, it is appropriate to split the dataset into train, test, and valid. However, looking at Figure 1 for test dataset, it is evident there is not a consistent number of photos for each breed. Since the goal is not to identify a specific human, but rather if the image is a dog or a human, splitting into the varying train, test, and valid are not necessary. These images are solely used to test the performance of the Human Face Detector.

## Solution Statement

The solution to this problem is defining a CNN that can determine the breed of a dog when supplied an image. If the image is a human, it will find a breed that most resembles the human. If the model believes with 100% certainty the dog is a pure bread, it will return the breed. It's interesting to note, if the dog is mixed (or if model is unable to return one specific dog), it will return the probability of the mix related to the breeds. The necessary steps to achieve this are broken down in the project design.

## Benchmark Model

From looking online at existing methods, it is determined that RESNET-50 provided the best results after some fine tuning – this is something that will be experimented more when working through the notebook. The provided notebook also stated there must be at least 10% test accuracy attained during the creation of a CNN to classify dog breeds from scratch. As well as 60% accuracy on the test set when creating a CNN to classify dog breeds using Transfer Learning.

## Evaluation Metrics

The project can be evaluated on effectiveness and efficiency based on the percentages provided above in the benchmark model. The notebook states the following: "[S]etting aside the fact that the classes are slightly imbalanced, a random guess will provide a correct answer roughly 1 in 133 times, which corresponds to an accuracy of less than 1%." Since the classes are imbalanced, the model will also be evaluated looking at the loss.

## Project Design

- Step 0: Import Libraries and Datasets
- Step 1: Detect Humans using OpenCV's implementation of Haar feature-based cascade classifier
- Step 2: Detect Dogs using pre-trained VGG-16 Model (as a starting point)
- Step 3: Create a CNN to Classify Dog Breeds (from Scratch)
- Step 4: Create a CNN to Classify Dog Breeds (using Transfer Learning) (RESNET-50)
- Step 5: Write an Algorithm that combines detected humans, detected dogs
- Step 6: Test Your Algorithm on points discussed in Problem Statement
    - an estimate of the dog's breed if a dog is detected
    - an estimate of the dog breed that is most resembling if a human is detected
    - state not a dog or human if neither a dog or human is detected
- Step 7: Deploy Model

## References

[1] https://www.rover-time.com/3-reasons-to-know-your-dogs-breed/

[2] https://dogell.com/en/compare-dog-breeds/beauceron-vs-rottweiler-vs-doberman-pinscher