

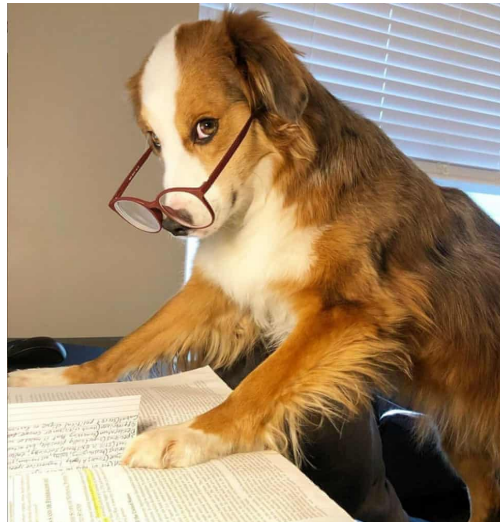
# Machine Learning Engineer Nanodegree

---

## Capstone Proposal

Francesco Paolo Albano 19/06/2020

[Project link](#)



## Domain Background

The dog breed classification from an image is very challenging problem even for a human eye. This is a multi class classification problem where we use supervised learning. With this project we will build a pipeline to identify from a single image, if a dog is present in image, and the its breed. The dataset of dog images is downloaded from [here](#). The dataset contains 8351 images of dog, divided into 133 dog breeds.

## Problem Statement

The goal of this project is to use the Convolutional Neural Networks and the Transfer Learning to create a pipeline that classifies the dog breeds. This pipeline will be inserted in a web app where user can take a picture and obtain prediction from this pipeline.

The pipeline is formed by:

- Human face detector that checks if there is a human face in the image.
- Dog detector that checks if there is a dog in the image.
- Breed dog detector that predicts the dog breed.

We want to achieve at least 70% of accuracy on dog breed prediction. We will use the transfer learning to build a CNN model with a good accuracy and less training time compared to creating and training a complex CNN from scratch.

## Datasets and Inputs

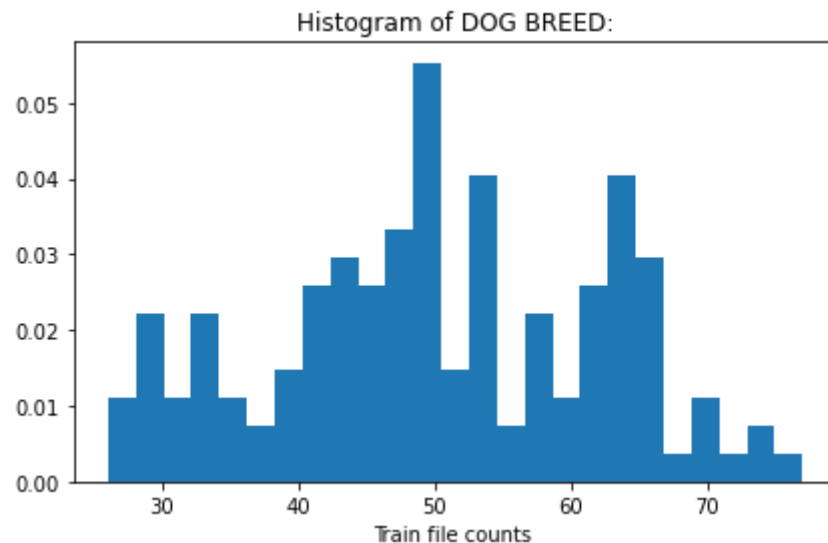
For this project we have 2 datasets provided from Udacity.

## Dog Dataset

This dataset, downloaded from [here](#), contains 8351 images of dog, divided into 133 dog breed. This dataset is already splitted in:

- *train*: 6680 images. (**80%**)
- *validation*: 835 images. (**10%**)
- *test*: 836 images. (**10%**)

In this histogram we can see the distribution of trainin images among the various dog breeds. With a mean of 50.23 images for dog breed.



We can see a small imbalancing between dog breed, we will try to reduce the proble using the data augmentation on train images using the (Transforms)

[<https://pytorch.org/docs/stable/torchvision/transforms.html>] of library torchvision.

## Human Dataset

This dataset, downloaded from [here](#). contains 13234 human images. All images have a shape of 250x250 pixel. We will use this dataset only for test a pre-built model for human face identifier.

## Solution Statement

### Dog breed classification

To perform the classification of dog breeds we have created from scratch a Convolutional Neural Network (without transfer Learning) with a simple architecture implementation. This model was composed by only 3 layer of CNN and achieve an accuracy of only 16%.

The model structure:

```
Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(2, 2))
Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
Linear(in_features=21632, out_features=1024, bias=True)
```

```
BatchNorm1d(1024, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
Linear(in_features=1024, out_features=133, bias=True)
```

The second model was created using transfer learning. Starting from a pretrained model of [ResNet18](#) downloaded from [torch pretrained models](#). I have choose the ResNet18 because have very good performance and relatively not very high complex architecture. Changing the last full connected layer and retraining only this last layer we have achieved a very good performance (**approximately 80%**) in only 5 epochs e few minutes of training.

## Pipeline steps

- The first step of pipeline is a dog detector based on VGG16 pretrained model that predict if is present a dog in the image.
- If a dog is found in the image, we will call the model, created using the transfer learning, to predict the dog breed.
- If there isn't dog in the image, we will check if is present an human in the image using the [OpenCv Haar Cascade classifier](#), and the we try to predict the most similar dog breed to human face.

## Benchmark Model

The test folder that contains 836 images of dog will be used as benchmark. A good model will achieve at least 70% of accuracy.

## Evalutaion Metrics

We will use the [Accuracy](#) metric to evaluate our models.

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

## Project Design

- Download the dataset and analyze distribution of images.
- Create a human face detector based on OpenCV Haar cascade classifier.
- Create a dog detector function based on VGG16 pretrained model.
- Create a first CNN model from scratch, and a CNN model using transfer learning.
- Create a function that from a sigle image check if is present a dog and then predict the breed.
- Create a simple web app to expose the model.

## Presentation

- ResNet Model: <https://arxiv.org/pdf/1512.03385.pdf>
- Torchvision pretrained models: <https://pytorch.org/docs/stable/torchvision/models.html>
- Transfer Learning for Computer Vision Tutorial: [https://pytorch.org/tutorials/beginner/transfer\\_learning\\_tutorial.html](https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html)
- OpenCV Haar Cascade Classifier: [https://docs.opencv.org/3.4/db/d28/tutorial\\_cascade\\_classifier.html](https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html)