Machine Learning Engineer Nanodegree

Capstone Proposal

Ivan Jennings
11 July 2021

Proposal

Create a dog breed classifier that can be deployed for us on a mobile or web app for educational purposes.

Domain Background

There are thousands of dog breeds all over the world many which are easily recognisable and also many variations of each breed. Given the number of variations in breeds, this can make it difficult to determine features to use - "*This domain is especially challenging since the appearance of corresponding parts can vary dramatically, e.g., the faces of bulldogs and beagles are very different.*"1 There are already existing classifiers available for use such as in the paper mentioned above1, however many of the applications are built for adult use. I would like to make the resulting classifier available for a younger audience to use as an educational tool.

Problem Statement

As previously mentioned, there are many different dog breeds and variations. This project aims to create an accurate classifier which will determine first if the photo contains a human and/or dog, and then determine the breed of the dog. The model will be deployed so that it can be used in a mobile or web app.

Datasets and Inputs

The data set that will be used for this project has been provided by Udacity and are available in the referenced links 2. The dog data set containing 8351 colour images which will be used for classifying breeds, contains 3 folders each for training, test and validation sets. Each set contains a folder of photos for each of the 133 dog breeds. The sets are not balanced between dog breeds - test sets have been 3-10 images for each breed, training sets have between 26-77 images per breed and the validation set has between 4-9 images per breed. The images also have varying resolutions in JPG format. We also have a human face data set with 13233 colour images which will be used to train a model for determining if a human face is present in a photo - this set has a range of people with about 1 image per person, with up to 40+ images per person.

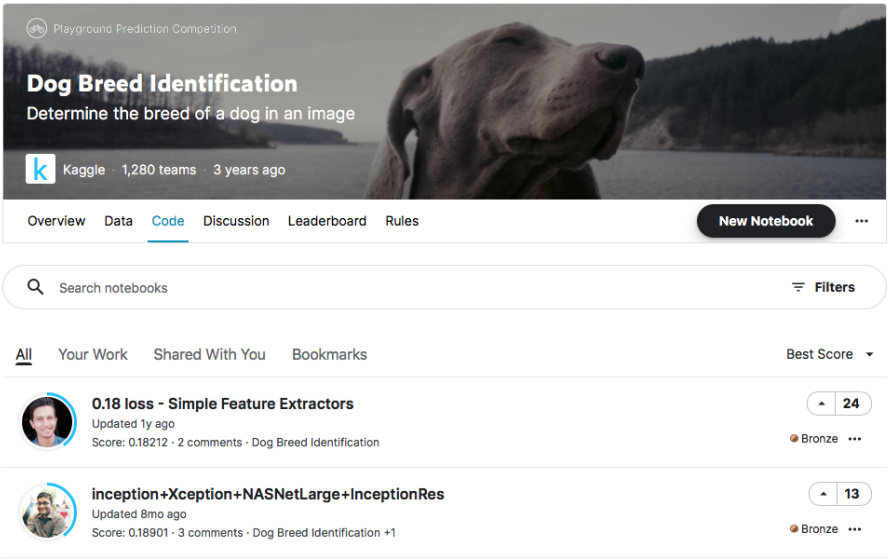
Solution Statement

I will use the provided data sets to create classifiers which determine if a target photo contains a human or dog. And also a CNN which will determine, with proposed accuracy mentioned in the evaluation metrics below, which breed the dog is. Later I will also use transfer learning to increase the accuracy of the solution.

Benchmark Model

The benchmark for the human and dog detector should be quite accurate - close to 100% correctly detected, however the differences between certain breeds can be quite subtle and thus more difficult to differentiate. The benchmark we will use for the classifier of breeds will be at least 10% for the initial CNN model written from scratch, and at least 60% accuracy for the CNN model using transfer learning.

I will also compare with similar models that can be found on Kaggle 3. The best model had a Multi Class Log Loss of 0.18 which we can use to compare our model.



Evaluation Metrics

I will calcuate accuracy of the detectors of human faces and dogs based on the total correctly detected images divided by the total predictions which should give us a result close to 100%. I will also calculate the accuracy of both the CNN models based on total correctly classified divided by the total classified. In addition to this I will use the loss calculation built into the scikit-learn library 4 to compare against the kaggle models.

Project Design

The following is the basic workflow that I'll be using, of which the template is provided by Udacity. I will also deploy the algorithm to be used on a basic web app. During the data preprocessing one of the challenges will be standardizing/augmenting the images, given that there are a range of images with different resolutions and angles. For the initial CNN model to classify dog breeds, I will start off with a simple mode from scratch using a small number of layers and epochs after which I can increase these to find an optimal model - I may use a tuning function to further optimize the model. For the transfer learning model, I will initially use the ResNet pre-trained model as this appears to be a popular one for this function after which I may try another model depending on the performance against the benchmarks.

Step 0: Import Datasets
Step 1: Image Preprocessing
Step 2: Detect Humans
Step 3: Detect Dogs
Step 4: Create a CNN to Classify Dog Breeds (from Scratch)
Step 5: Create a CNN to Classify Dog Breeds (using Transfer Learning)
Step 6: Write your Algorithm
Step 7: Test Your Algorithm
Step 8: Deploy Algorithm using a Lambda function & RESTful API
Step 9: Create basic web app to upload photo and show detected information

References

1
J. Liu, A. Kanazawa, D. Jacobs, and P. Belhumeur, "Dog Breed Classification Using Part Localization", Computer Vision–ECCV 2012. Springer Berlin Heidelberg, 2012. 172-185.

2
Dog dataset - Udacity
Human dataset - Udacity

3
Dog breed identification

4
Scikit-learn loss function