Capstone Project
Machine Learning Engineer
Nanodegree

Abdoulaye Beavogui Koma
June 18, 2020

Definition

Project Overview

According to the American Society for the Prevention of Cruelty to Animals, approximately 6.5 million companion animals enter U.S. animal shelters nationwide every year. Of those, approximately 3.3 million are dogs [1], which accounts for roughly 50.8% of the population. During the onboarding process of these dogs, it is critical to identify the breed in order to provide the most effective treatment early. With the accelerated pace in digital image processing, artificial intelligence and computer vision, we can accelerate the dog breed identification process so that dog caretakers can focus on the most important aspect: taking care of sheltered dogs.

As such, we will be working on the dog breed classifier project given by Udacity, Inc. as a first step to tackle this real problem. In this project, we will build an application that can process images containing dogs and identify the corresponding dog breeds.

Problem Statement

The goal of this project is to create a dog breed identification application. This application will detect if there is a human or a dog in the image and estimate the corresponding dog breed. In the case that only a human is identified, the application will estimate the dog breed he/she resemble the most.

Data Exploration

The dataset provided includes 21,584 colored images in a JPEG format:

- Dog dataset contains 8351 images parsed in 133 folders, each corresponding to a unique dog breed. The images come in all shape and sizes and some images may contain multiple dogs or a human with a dog. The dataset is imbalanced as we see can an uneven class distribution of dog breed images. The plot below shows the image count available per dog breed, where 9 out of 133 breeds contain more than 124 images (above one standard deviation away from the mean)

[1] "Pet Statistics: How many pets are in the United States? How many animals are in shelters?" https://www.aspca.org/animal-homelessness/shelter-intake-and-surrender/pet-statistics

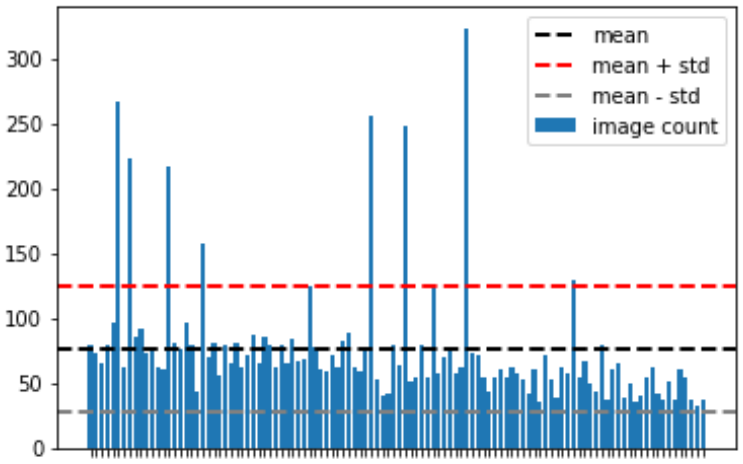

Fig1. Class distribution of dog breed image count per dog breed

- Human dataset contains 13233 images of humans. All images are of size 250x250 and are center cropped to the human face

Solution Statement

The application will be divided in three main modules: a human detector, a dog detector and a dog breed classifier.

- Human detector: its purpose is to identify a human in a given image. We will look at several OpenCV face detectors to identify the appropriate model to use

- Dog detector: its purpose is to identify a dog within a given image. We will look at pretrained Convoluted Neural Network (CNN) models to identify the appropriate model to use

- Dog breed classifier: its purpose is to estimate the most resembling dog breed in a given image. We will build a CNN (either from scratch or a pretrained model) to estimate the dog breed in a given image

The application will perform in the following order:
=> read an image => identify a dog in the image => identify a human in the image => provide an estimate of dog breed in the image => plot the image and provide result

Fig2. Example of the application output

If the application cannot identify a human or a dog in the image, the application will report that it's neither a dog nor a human and will not provide a dog breed estimation.

Benchmark and Metrics

We have the following metrics to consider for evaluating the performance and quality of the application:

- Response time: Time it takes for a particular set of instructions to run. We'll be looking at how long it takes to run different modules and models to identify opportunities for optimization

$$Response\ time = end\ time - start\ time$$

- Accuracy of the detectors: Number of true positives and true negatives (or correct identifications) over the entire dataset. This metric will tell us how well our application detects humans, dogs and estimates the corresponding dog breed

$$Accuracy = \frac{true\ positives + true\ negatives}{dataset\ size}$$

- Precision of the detectors (optional): Number of true positives over all positives. A lower number of false positives will lead to a higher precision

$$Precision = \frac{true\ positives}{true\ positives + false\ positives}$$

- Recall (or sensitivity) of the detectors (optional): Number of true positives over the sum of true positives and false negatives. A lower number of false negatives will lead to a higher recall

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

- F1 score of the detectors (optional): A metric that considers both precision and recall (harmonic mean). It is relevant when provided an unbalanced class distribution. The closer to 1, the more precise and accurate the application is

$$F1\ score = 2 \times \frac{Precision\ \times\ Recall}{Precision + Recall}$$

There are many widely popular open-source models available to detect humans, dogs and dog breeds in a given image higher than 80%. As an example, a team from the Istanbul Technical Institute provided the following accuracy of their dog breed classifier models: ResNet-50 (89.66%), DenseNet-121 (85.37%), DenseNet-169 (84.01%) and GoogleNet (82.08%) [2].

Our goal is to achieve at least an 80% accuracy on a test dataset. We will benchmark our models (more specifically the dog breed classifier: CNN scratch vs CNN pretrained) in terms of performance and accuracy to identify the best approach in designing the application.

Project Design

To complete this project, we will break down the project into a series of steps:

1. Explore and process data:
    - Import, parse and explore the dataset
    - Transform dataset for machine learning

2. Modeling:
    - Model human detector:
        - Identify the different model frameworks for human detection
        - Benchmark models and select best model for application
    - Model dog detector:
        - Identify the different model frameworks for dog detection
        - Benchmark models and select best model for application
    - Model dog breed estimator
        - Identify the different model framework for dog breed estimation (CNN)
        - Train, evaluate models and select best model for application
    - Build application:
        - Integrate all the models to build a seamless application

3. Deployment:
    - Run application:
        - Run application on real images and observe performance

[2] "Modified Deep Neural Networks for Dog Breeds Identification" https://www.researchgate.net/publication/325384896_Modified_Deep_Neural_Networks_for_Dog_Breeds_Identification

The project template is available through this GitHub link provided by Udacity, Inc. The solution will be developed using Udacity, Inc. workspace environment (with GPU available). In the event that training and running a model takes too much time, we will leverage Amazon Web Services (AWS) SageMaker as an alternative environment to build, train and test the models required for the application.