Capstone Proposal Bipin Sawant

Machine Learning Engineer Nanodegree June 11, 2020

Definition

Domain Background

The Dog breed classifier is a well-known problem in ML community. The problem is to identify a breed of dog if dog image is given as input, if supplied an image of a human, we must identify the resembling dog breed. The idea is to build a pipeline that can process real world user supplied images and identify an estimate of the canine's breed. This is a multi-class classification problem where we can use supervised machine learning to solve this problem. After completing this model, I am planning to build a web app where user can input an image and obtain prediction from this model. This project gives me an opportunity to build and deploy ML models, so I have chosen this as my capstone project.

Problem statement

The goal of the project is to build a machine learning model that can be used within web app to process real-world, user-supplied images.
The algorithm must perform two tasks:

- **Dog face detector:** Given an image of a dog, the algorithm will identify an estimate of the canine's breed.
- **Human face detector:** If supplied an image of a human, the code will identify the resembling dog breed.

Datasets and inputs

To solve the problem our input data must be images because we want the users to take an image of a dog (or human) with their phone, sent it to our server and we would identify breed of the dog.

All data for this project is provided by Udacity. We have pictures of dogs and pictures of humans.

All dog pictures are sorted in train (~6,680 Images), test (~836 Images) and valid (~835 Images) directory, and all the images in these directories are sorted in breed

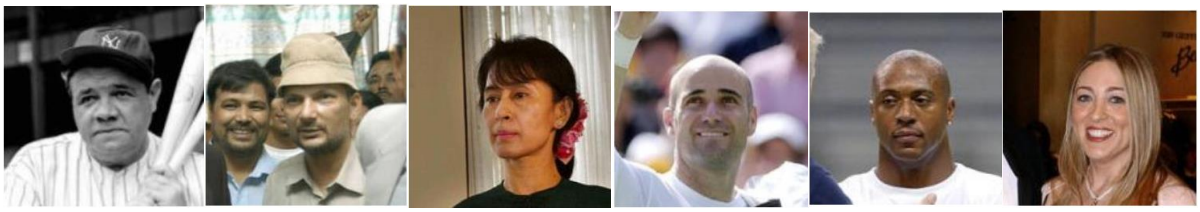
directories. We have ~133 folders (dog breeds) in every train, test and valid directory.

Human pictures are sorted by name of each human. We have ~13,234 Files (Images), ~5,749 Folders (Humans).

Our data is not balanced because we have one image of some people and several for others. The same is for dog images. (the difference is from 1 to 9 images in most cases).

Dog images have different image sizes, different backgrounds, some dogs are in full sizes and some just ahead. Lightning is not the same. That is okay because we don't know how users' images will be, and we want that our model works on different types of images. Human images are all the same size 250×250. Images are with different backgrounds, light, from different angles, sometimes with few faces on the image.

Some of the sample images



Solution statement

I will use Convolutional Neural Networks (CNN) to solve the problem. CNN is a part of deep neural networks and is great for analyzing images. The solution involves following steps

- detect human images, we can use existing algorithm like OpenCV's
- detect dog-images, we will use a pretrained VGG16 model.
- after the image is identified as dog image or a human image, we can pass this image to a CNN that will process the image and predict the closest match of dog breed.

Benchmark model

The CNN model created from scratch must have accuracy of at least 10%. This can confirm that the model is working because a random guess will provide a correct answer approximaltely 1 in 133 times, which corresponds to an accuracy of less than 1%.

The CNN model created using transfer learning must have accuracy of 60% and above.

Evaluation metrics

For this multi class classification, Multi class log loss will be used to evaluate the model. Because of the imbalance in the dataset, accuracy is a not a good indicator here to measure the performance. Log loss takes into the account of uncertainty of prediction based on how much it varies from actual label and this will help in evaluating the model.

Project design

Step 1: Import the necessary dataset and libraries, Pre-process the data and create train, test and validation dataset. Perform Image augmentation on training data.

Step 2: Detect human faces using OpenCV.

Step 3: Create dog detector using pretrained VGG16 model.

Step 4: Create a CNN to classify dog breeds from scratch, train, validate and test the model.

Step 5: Create a CNN to Classify Dog Breeds using Transfer Learning with resnet101 architecture. Train and test the model.

Step 6: Write an algorithm to combine Dog detector and human detector.

- If dog is detected in the image, return the predicted breed.
- If human is detected in the image, return the resembling dog breed.
- If neither is detected, provide output that indicates the error.