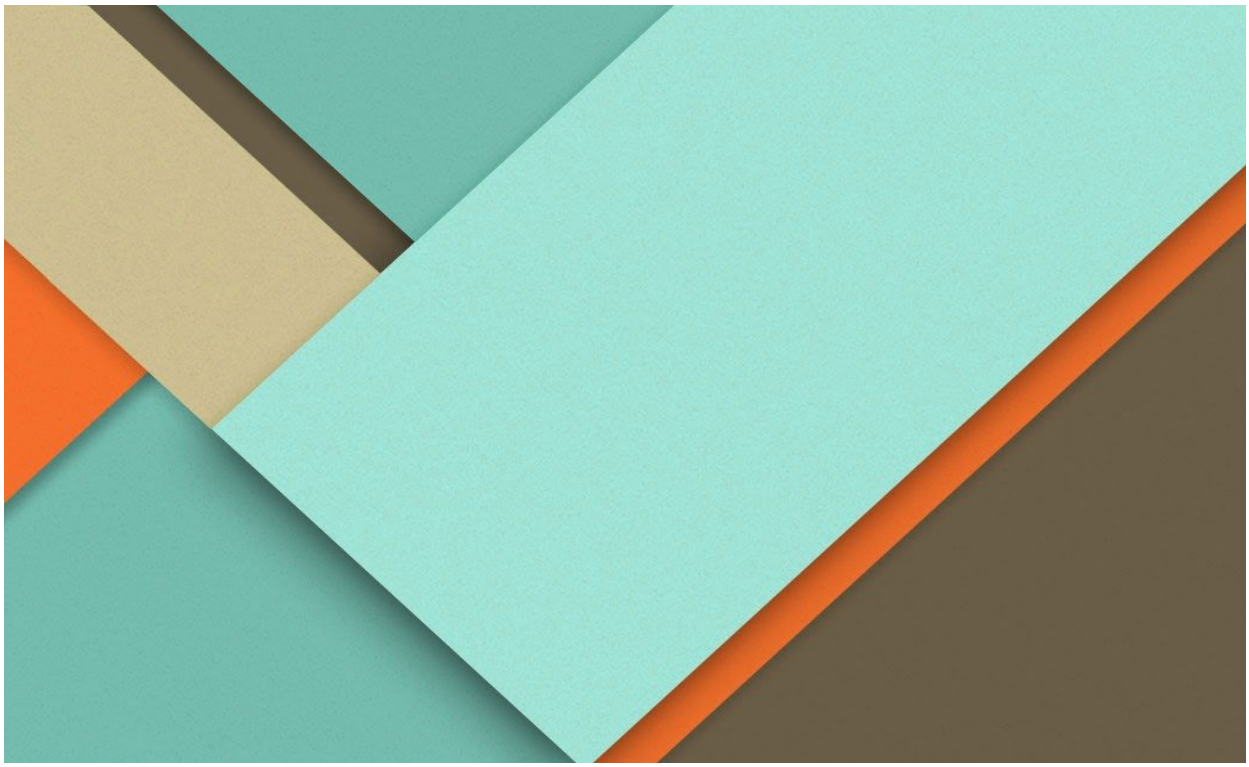

Dog Breed Classifier Capstone Proposal

17-10-2020

Ahmed Gharib

Machine Learning Engineer Nanodegree

Udacity

Domain Background

There are several breeds that most people have never seen before, or even heard of. Some are newly registered, and some are just less common in the United States.

Some of these dogs are (Beauceron, Canaan Dog, Cesky Terrier, Chinook, Dandie Dinmont Terrier and Lagotto Romagnolo).

In the United States alone, the AKC's dog breed list currently includes 190 dog breeds. Worldwide, the FCI lists 360 officially recognized breeds. These don't include experimental breeds that have yet to achieve official status. Official lists also don't include mixed-breed dogs, not even "designer" crossbreeds like the goldendoodle (a cross between a golden retriever and a poodle) or the puggle (a mix of beagle and pug).

But on the other hand some people in the world have no clue about any of the dog breeds and can’t even tell the difference between them.

Problem Statement

It’s an Image Classification problem where we need to classify images of dogs and define their breeds and if the image is for a human we classify it as a human and give the closest dog breed for this human image.

Datasets and Inputs

Datasets: the datasets are provided by udacity as a 2 zipped folders contain:

1. Human Dataset:

LFW folder with total 13233 images of humans distributed in 5749 folders each folder contains images for one person but there is variation in the number of images for each folder.

2. Dog Dataset:

Distributed in 3 main folders with a total of 8351 images.

1. Train: 6680 images 80%
2. Valid: 835 images 10%
3. Test: 836 images 10%

Each folder contains 133 folders for each dog breed but also here the images are not equally distributed for each class or breed of dogs.

Solution Statement

We will need here to create 2 models

1. A model to predict if there is a human in the picture.
2. A model to predict the dog breed of a given picture.

And then Write an algorithm that accepts a file path to an image and first determines whether the image contains a human, dog, or neither. Then,

1. if a dog is detected in the image, return the predicted breed.
2. if a human is detected in the image, return the resembling dog breed.
3. if neither is detected in the image, provide output that indicates an error.

Benchmark Model

1. A Model created from scratch with a minimum accuracy of 10% that confirms that the model is working better than a random selection as we have 133 classes and a random model will perform an accuracy of less than 1%.
2. A Model created using transfer learning with a minimum accuracy of 60%

Evaluation Metrics

Accuracy : comparing the predicted labels for our images with the true labels and compute the percentage of our model accuracy as follow

(Number of correctly predicted images/ Total Number of images) *100

Project Design

1. Import Datasets

Download the dog dataset. Unzip the folder and place it in this project's home directory, at the location /dogImages.

Download the human dataset. Unzip the folder and place it in the home directory, at location /lfw.

Then perform image preprocessing before we input them to the models.

2. Detect Humans

In this section, we use OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces in images.

3. Detect Dogs

In this section, we use a pretrained VGG-16 model to detect dogs in images.

4. Create a CNN to Classify Dog Breeds (from Scratch)

Now that we have functions for detecting humans and dogs in images, we need a way to predict breed from images. In this step, we will create a CNN that classifies dog breeds.

5. Create a CNN to Classify Dog Breeds (using Transfer Learning)

We will now use transfer learning to create a CNN that can identify dog breed from images. Our CNN must attain at least 60% accuracy on the test set.

6. Write the Algorithm

Write an algorithm that accepts a file path to an image and first determines whether the image contains a human, dog, or neither. Then,

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

7. Testing the Algorithm

In this section, we will take the new algorithm for a spin! What kind of dog does the algorithm think that *you* look like? If you have a dog, does it predict your dog's breed accurately? If you have a cat, does it mistakenly think that your cat is a dog?

References

1. Deep Learning Specialization at Coursera
2. PyTorch documentation
3. Intro to Deep Learning with PyTorch Free Course at Udacity
4. 10 dog breeds probably never hear of (Article)
5. How many dog breeds are there (Article)