Machine Learning Engineer Capstone Proposal

Mendy Wu 08/11/2021

Domain Background

People see a dog and wonder it's breed. Also, some owners are wondering their dogs' breed. Without understand the breed of their dogs, their dog is at risk for a number of problems that come from breeding. Finding the answer by google the descriptions of the dog is difficult and time consuming. However, the process will be much easier if they can submit an image of the dog and get an answer via a web or mobile app.

Problem Statement

In this project, I will create a web application that can distinguish an image of dog from human images and classify the breed of the dog.

Datasets and inputs

Users would like take images of the dog and ask for the breed. Therefore, we need to feed images as input for the models. The data sources are showing below:

1. dog dataset: [Link]
2. human dataset: [Link]

There are 13233 colored human images with size 250*250. in the human dataset. While there are 8351 dog images with various image sizes. All the dog images are colored with the heights from 113 to 4003 and the weights from 105 to 4278. To make a better training and prediction, it would be better to transform the dog images with same size before feeding into the model.

The datasets including 3 parts: the training, validation, and testing datasets. All the datasets will be used for training, valid and testing the model to get the best model with low losses.

The human images will use for the model identifies the object in the image is a human or a dog. While the dogs in the images have different breeds, backgrounds, postures, and orientations. These dogs images will feed into another model predict the dog’s breed.

Solution Statement

I will create the application by following steps:

1. Train a classifier that can determine human faces in images by using OpenCV's Haar feature-based cascade classifier

The Haar feature-based cascade classifiers is one of the pre-trained face detectors in the OpenCV. We need to convert the loaded BFR images to grayscale by using the function cvtColor first. And then use the grayscale image as the parameter. Adding the parameter in the function detectMultiScale to detect the faces in the image. The steps will be written in a function called face_detector and test the images of human and dogs.

2. Build a classifier that can detect the image of a dog by using pre-trained VGG-16 model
3. Create a simple CNN model from scratch to classify dog breeds
4. Create a CNN model by using a pre-trained model to classify dog breeds
5. Deploy the model with a web application runs on the server

Benchmark Model

I will use the simple CNN model as my benchmark model and compare the accuracy of the final model

Evaluation Metrics

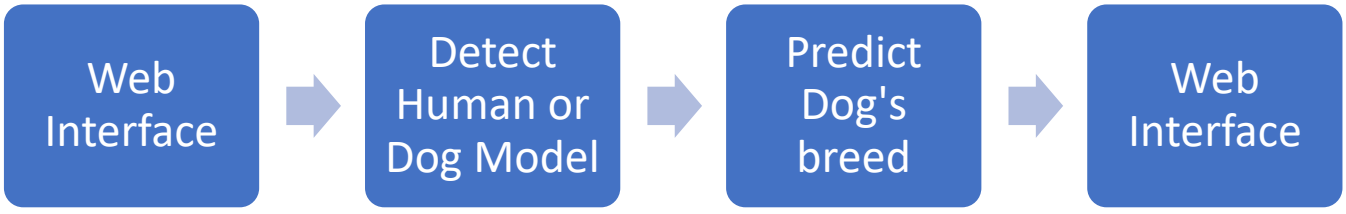
I will use accuracy as the evaluation metrics. The expected accuracy of different classifiers are:

1. The expected accuracy of detecting human face is at least 90%
2. The accuracy of predict dog breed by using the scratch model needs to be greater than 10%
3. The accuracy of predict dog breed by using the pre-trained model needs to be at least 60% on the test data

Project Design

The code will be in an iPython notebook, and a report with the technology details will be present. Both will keep in a Github repository with an instruction of how to implement the web application in the README file.

The workflow of the model shows below:



The image will load via the web interface. The uploaded image will feed into the human or dog model first to detect if the image contains a dog. If the image contains a dog, the image will feed into the dog's breed model. Otherwise, "this is a human" will return to the web interface as a result. For these dog

images, the dog’s breed model will predict the breed of the dog and return the results on the web interface.