Capstone Project

Mohammed Azeem
26th June, 2020

Machine Learning Engineer Nanodegree

Dog Breed Classifier using Convolutional Neural Network

Domain Background

Dog breeding is a known problem in ML. The problem is identifying a dog breed if a dog's image has been assigned as an input, and when providing a human image, it should identify the same type as the one made for the dog. The idea is to create a pipeline that can process real-world images provided by the user and identify canine-type equations. This is a multi-level problem where we can use machine learning to be monitored to solve this problem. After completing this model, I plan to build a web app where the user can upload an image and receive predictions from the model. This project gives me the opportunity to create and export ML models, so I chose this as my capstone project

Problem Statement

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

Dog face detector: Given an image of a dog, the algorithm tries to estimate the breed of the dog.

Human face detector: When an image of a human is supplied , the code will identify the closest resembling dog breed.

Datasets and Inputs

For this project, the input format should be of the image type, because we want to insert the image and identify the dog type. Details of the data for this project were provided by Udacity. Data with pictures of dogs and people.

Dog Photos Data: The dog data database contains 8351 images sorted by train (6,680 Photos), test (836 photos) and valid directions (835). Each of these directories (train, test, active) has 133 folders corresponding to dog breeds. Images are of different sizes and backgrounds; some images are not full-size. The details are not the same because the number of photos provided per genre varies. Few have 4 pictures and some have 8 pictures.

Demographic data: Demographic data contains 13233 demographic images organized into individual names (5750 folders). All images are 250x250 in size. Images have different domains and different angles. Details are not the same because we have 1 photo for some people and lots of photos for others.

Solution Statement

By performing this multiclass separation, we can use Convolutional Neural Network to troubleshoot. The Convolutional Neural Network (CNN) is an in-depth learning algorithm that can capture input images, assign value (readable bits and selections) to various objects / objects in an image and be able to distinguish one from another. The solution involves three steps. First of all, to get human photos, we can use an existing algorithm such as OpenCV's Haar Phase implementation which is supported by large. Secondly, to find dog photos we will use the VGG16 model that has been developed. Finally, after the image has been identified as a dog / human being, we can transmit the image to CNN that will process the image and predict the best possible breeding of the 13 breeding methods.

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 roughly 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

In this classification, multiple missing values will be used to evaluate the model. Due to differences in the data set, accuracy is not a good indicator here to measure performance. Log loss takes account of the uncertainty of the prediction based on how different it is from the original label and this will help to evaluate 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's implementation of Haar feature based cascade classifiers.

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.