Capstone Project

Anup Kolte

Machine Learning Engineer Nanodegree

May 19, 2020

Dog Breed Classifier using CNN

Domain Background

The Dog breed classifier could be a well-known problem in ML. the matter is to spot a breed of dog if dog image is given as input, if supplied a picture of a person's, we've to spot the resembling dog breed. the thought is to create a pipeline that may process planet user supplied images and identify an estimate of the canine’s breed. this can be a multi-class classification problem where we are able to use supervised machine learning to resolve this problem. After completing this model, i'm going to build an internet app where user can input a picture and procure prediction from this model. This project gives me a chance to create and deploy ML models, so i've got chosen this as my capstone project.

Problem Statement

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

Dog face detector: Given a picture of a dog, the algorithm will identify an estimate of the canine’s breed.

Human face detector: If supplied a picture of a person's, the code will identify the resembling dog breed.

Datasets and Inputs

For this project, the input format must be of image type, because we wish to input a picture and identify the breed of the dog. The dataset for this project is provided by Udacity. The dataset has pictures of dogs and humans.

Dog images dataset: The dog image dataset has 8351 total images which are sorted into train (6,680 Images), test (836 Images) and valid (835 Images) directories. Eachof this directory (train, test, valid) have 133 folders appreciate dog breeds. the photographs are of various sizes and different backgrounds, some images don't seem to be full-sized. the information isn't balanced because the quantity of images provided for every breed varies. Few have 4 images while some have 8 images.

Human images dataset: The human dataset contains 13233 total human images which are sorted by names of human (5750 folders). All images are of size 250x250. Images have different background and different angles. the information isn't balanced because we've 1 image for a few people and plenty of images for a few.

Solution Statement

For performing this multiclass classification, we are able to use Convolutional Neural Network to resolve the matter. A Convolutional Neural Network (CNN) could be a Deep Learning algorithm which may absorb an input image, assign importance (learnable weights and biases) to numerous aspects/objects within the image and be ready to differentiate one from the opposite. the answer involves three steps. First, to detect human images, we are able to use existing algorithm like OpenCV’s implementation of Haar feature based cascade classifiers. Second, to detect dog-images we are going to use a pretrained VGG16 model. Finally, after the image is identified as dog/human, we are able to pass this image to an CNN which is able to process the image and predict the breed that matches the simplest out of 133 breeds.

Benchmark Model

- The CNN model created from scratch must have accuracy of a minimum of 10%. this may confirm that the model is functioning because a random guess will provide an accurate answer roughly 1 in 133 times, which corresponds to an accuracy of but 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 are going to be accustomed evaluate the model. thanks to the imbalance within the dataset, accuracy could be a not a decent indicator here to live the performance. Log loss takes into the account of uncertainty ofprediction supported what quantity it varies from actual label and this may help in evaluating the model.

Project Design

Step 1: Import the required dataset and libraries, Pre-process the information and make 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 mix Dog detector and human detector.

- If dog is detected within the image, return the expected breed.

- If human is detected within the image, return the resembling dog breed.

- If neither is detected, provide output that indicates the error.