Machine Learning Engineer Nanodegree

Capstone Project

Dog Breed Classifier using CNN

Parul Agarwal

8th August,2020

Domain Background

The Dog breed classifier is a well-known problem in ML. The problem is to identify a breed of dog if dog image is given as input, if supplied an image of a human, we have to 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 objective of the venture is to assemble an machine learning model that can be utilized inside web application to process genuine world, client provided pictures. The calculation needs to perform two undertakings:

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

For this task, the information group must be of picture type, since we need to enter a picture and distinguish the variety of the canine. The dataset for this task is given by Udacity. The dataset has pictures of canines and people.

Canine images dataset: The canine picture dataset has 8351 all out pictures which are arranged into train (6,680 Images), test (836 Images) and substantial (835 Images) registries. Each of this registry (train, test, substantial) have 133 organizers relating to canine varieties. The pictures are of various sizes and various foundations, a few pictures are most certainly not full-sized. The information isn't adjusted on the grounds that the quantity of pictures accommodated each breed changes. Few have 4 pictures while some have 8 pictures.

Human images dataset: The human dataset contains 13233 complete human pictures which are arranged by names of human (5750 organizers). All pictures are of size 250x250. Pictures have diverse foundation and various edges. The information isn't adjusted since we have 1 picture for certain individuals and numerous pictures for a few.

Solution Statement

For playing out this multiclass grouping, we can utilize Convolutional Neural System to take care of the issue. A Convolutional Neural Network (CNN) is a Deep Learning calculation which can take in an info picture, dole out significance (learnable loads and inclinations) to different angles/objects in the picture and have the option to separate one from the other. The arrangement includes three stages. To start with, to distinguish human pictures, we can utilize existing calculation like OpenCV's usage of Haar highlight based course classifiers. Second, to recognize canine pictures we will utilize a pretrained VGG16 model. At last, after the picture is distinguished as canine/human, we can pass this picture to a CNN which will process the picture and anticipate the variety that coordinates the best out of 133 varieties.

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

For this multi class classification, Multi class log loss will be used to evaluate the model. On account of the irregularity in the dataset, accuracy is a not a good indicator here to gauge 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'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.

References

1. Original repo for Project - GitHub:
https://github.com/udacity/deep-learning-v2-pytorch/blob/master/project-dog-classification/

2. Resnet101:
https://pytorch.org/docs/stable/_modules/torchvision/models/resnet.html#resnet101

3. Imagenet training in Pytorch:
https://github.com/pytorch/examples/blob/97304e232807082c2e7b54c597615dc0ad8f6173/imagenet/main.py#L197-L198

4. Pytorch Documentation: https://pytorch.org/docs/master/

5.
https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neuralnetworks-the-eli5-way-3bd2b1164a53

6. http://wiki.fast.ai/index.php/Log_Loss