Capstone Project

Dhanush M

Machine Learning Engineer Nanodegree

May 23rd 2020

Dog Breed Classifier Using CNN

Domain Background

In the last decade, machine learning (ML) has become more popular thanks to very powerful computers that can handle lots of data in a reasonable amount of time. Machine learning concept was introduced by Arthur Samuel in 1959 so it is not new, but today we can use lots of its potential.

The Dog breed classifier is a well-known problem in ML.After completing this model, I am planning to build a web app where users 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 problem is to identify a breed of dog if a dog image is given as input, if supplied an image of a human, we have to identify the resembling dog breed.

This is a multi-class classification problem where we can use supervised machine learning to solve this problem.

Datasets and Inputs

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

All dog pictures are sorted in train(6,680 Images), test(836 Images) and valid(835 Images) directory, and all the images in these directories are sorted in breed directories. We have 133 folders (dog breeds) in every train, test and valid directory.The images have different sizes.

Human pictures are sorted by the name of each human. We have 13,233 Files (Images), 5,750 Folders(Humans).All images are of size 250x250.

Solution Statement

We will use Convolutional Neural Networks (CNN) to make a model. CNN is a part of deep neural networks and is great for analyzing images.The solution involves three steps. First, to detect human images, we can use existing algorithms like OpenCV's implementation of Haar feature based cascade classifiers. Second, to detect dog-images we will use a pretrained VGG16 model. Finally, after the image is identified as dog/human, we can pass this image to a CNN which will process the image and predict the breed that matches the best out of 133 breeds.

Benchmark Model

- we will use the Convolutional Neural Networks (CNN) model created from scratch with an accuracy of more than 10%. This should be enough to confirm that our model is working because random guess would be 1 in 133 breeds which are less than 1% if we don't consider unbalanced data for our dog images.
- The CNN model created using transfer learning must have accuracy of 60% and above.

Evaluation Metrics

The problem we try to solve is a classification problem. Because our data is an unbalanced, simple accuracy score is not very good here. For this multi class classification, Multi class log loss will be used to evaluate the model. Because of the imbalance in the dataset, accuracy is not a good indicator here to measure the

performance. Log loss takes into account the uncertainty of prediction based on how much it varies from actual label and this will help in evaluating the model.

Project Design

After getting a dataset that is provided by Udacity, Pre-process the data and create train, test and validation dataset. Perform Image augmentation on training data.Then, detect human faces using OpenCV's implementation of Haar feature based cascade classifiers and create a dog detector using pretrained VGG16 model.Create a CNN to classify dog breeds from scratch, train, validate and test the model to get higher than 10% accuracy. Finally, Create a CNN to Classify Dog Breeds using Transfer Learning with resnet101 architecture. We will then train and test our model with the same test data as before but now we expect our accuracy to be over 60%.

We will then write an algorithm to combine dog detector and human detector such that 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 and if neither is detected, provide output that indicates the error.

Reference

1. Original repo of the Project - GitHub: https://github.com/udacity/deep-learning-v2-pytorch/blob/master/project-dog-classification/
2. Pytorch Documentation: https://pytorch.org/docs/master/
3. Imagenet training in Pytorch: https://github.com/pytorch/examples/blob/97304e232807082c2e7b54c597615dc0ad8f6173/imagenet/main.py#L197-L198