Capstone Project: Machine Learning Engineer Nanodegree

Dog Breed Classifier using CNN

Domain Background

Image classification is one of the fundamental problems in field of Machine Learning and Deep Learning. This forms core of several computer vision and real-world image processing problems. The objective is to accurately predict a breed of dog given an image as an input. If input is an image of a human being then the classifier to predict the label that closely resembles a canine breed of the dog. This model helps me to create classifier that can be deployed and integrated any web or mobile application to classify real world images.

Problem Statement

The primary goal of this project is to create a multi-class classifier using CNN. The trained model will predict a score that indicates the probability of an input image belonging to a specific breed of a dog. The project includes 2 main tasks:

1. Create a Dog detector that will predict the estimator of a canine's breed.
2. Create a human face detector that predicts resembling breed of the dog.

Datasets and Input

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.

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. Each of this directory (train, test, valid) have 133 folders corresponding to dog breeds. The data is not balanced because the number of images provided for each breed varies.

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. This dataset is also not balanced.

Solution Statement

For solving this multiclass problem, I will be using Convolutional Neural Network (CNN) which is a state-of-the-art algorithm for image classification problem. The approach I will be following includes 3 steps:

1. Detect human image using OpenCV which in turn uses a haar cascade classifier.
2. Detect dog image using a pretrained CNN model.
3. Finally, use our custom model to predict breed of the dog

Benchmark Model

The CNN model created from scratch must have accuracy of at least 60%. The CNN model created using transfer learning must have accuracy of 75% and above.

Evaluation Metrics

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 a not a good indicator here to measure 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.

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 custom CNN to classify dog breeds from scratch, train, validate and test the model from scratch.

Step 5: Create a CNN to Classify Dog Breeds using Transfer Learning technique.

Step 6: Write an algorithm to combine Dog detector and human detector.

- If dog is detected returns its predicted breed.
- If human is detected return the resembling breed of the dog.
- Else throw error as input image is not valid.

References

1. Original repo for Project - GitHub: https://github.com/udacity/deep-learning-v2pytorch/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