Proposal: Dog Breed Image Classification with CNN

Jason Zeng

July 7, 2020

Project Overview

This project aims to identify the breed of dog. Given an image of a dog, our algorithm will identify an estimate of the canine's breed. If supplied an image of a human, the algorithm will identify the resembling dog breed. The idea is to build a pipeline to process real-world, user-supplied images. This is a multi-class classification problem and we can use supervised learning to tackle it.

Domain Background

This project is derived from Computer Vision.

Problem Statement

The goal of this project is to build a machine learning model that can be used as a web app to process real-word, user-supplied images. It requires the implementation of two kinds of detections. First, **Dog detector**, which can identify the dog's breed when an image of a dog is given. Second, **Human detector**, which will identify the resembling dog breed if an image of a human is supplied.

Datasets and Inputs

The dataset for this project is provided by Udacity. They are pictures of dogs and humans. To be more specific, **Dog images dataset** has 8351 images in total which are split into train (6,680 images), test (836 images) and valid (835 images) directories. Each of this directory has 133 folders corresponding to dog breeds. These images have different sizes, different quality, and are imbalanced for different breeds. On contrary, **Human images dataset** has 13233 images in total and are stored in 5750 folders based on human names. Though these human images have different backgrounds and are also imbalanced, they are of the same size 250x250 pixels.

Solution Statement

We can use Convolutional Neural Networks (CNN) for this kind of multi-class classification tasks, by feeding images into the algorithm. The CNN architecture then learns knowledge (weights w

and bias b) by minimizing the difference between CNN's prediction with provided ground truth (labels). This minimization process is done through back propagation.

In this project, we will first use existing algorithms like OpenCV's implementation of Haar feature based cascade classifiers, to detect human images. Next, we will use a pretrained VGG16 model to detect dog images. Finally, after the image is identified as dog or human, we can pass this image to an CNN to predict the breed out of the 133 possible breeds.

Benchmark Model

- The CNN model created from scratch should have accuracy at least ~10% since a random guess has accuracy < 1% (1 out of 133 breeds).
- The CNN model created using transfer learning should have a much higher accuracy, say 70%.

Evaluation Metrics

For multi-class classification, multi-class log loss is a typical metric to be used to evaluate model performance. Accuracy is also used to get a feeling of overall model performance.

Project Design

Step 1: Import datasets and libraries

Step 2: Pre-processing. This includes train/test/validation split and image augmentations on training.

Step 3: Build Human image detector using OpenCV's Haar feature based cascade classifiers.

Step 4: Build Dog image detector using pretrained VGG16 model

Step 5: Create from scratch a CNN architecture to classify dog breeds. Train and test the model.

Step 6: Create a CNN architecture using Transfer Learning with ResNet101. Train and test the model.

Step 7: Develop an algorithm to combine Dog detector and Human detector.

- 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.
- If neither is detected, flag an error.

References

1. Original project repo on GitHub from Udacity: https://github.com/udacity/deep-learning-v2-pytorch/tree/master/project-dog-classification
2. Resnet101: https://github.com/KaimingHe/deep-residual-networks
3. Pytorch: https://pytorch.org/