Machine Learning Engineer Nanodegree

Capstone Proposal

Ikechukwu Nigel Ogbuchi
January 19, 2020.

Proposal: Dog Breed Classifier

Domain Background

Convolutional neural networks (CNN) have been used to achieve amazing results in various applications like image classification, face recognition, scene labelling and document analysis. It comprises a deep learning architecture which is inspired by the structure of visual system. Convolutional networks are trainable multistage architectures with each stage consisting of multiple layers. The input and output of each stage are sets of arrays called as feature maps. In the case of a colored image, each feature map would be a 2D array containing a color channel of the input image, a 3D array for a video and a 1D array for an audio input. The output stage represents features extracted from all locations on the input. Each stage generally consists of a convolution layer, non-linearity and a pooling layer. A single or multiple fully-connected layers are present after several convolution and pooling layers.

Problem Statement

The way image classification works is this: A trained model is given an input and it is able to predict a class (for example "This is a dog", "This is not a dog") as an output. It can also give you the probability that the input belongs to a particular class (for example: "This is a dog. 90% confidence!"). The goal of the project was to create a pipeline that takes an image and detects whether a human or dog is present, predicting the breed for the dog or deciding what dog breed the human looks similar to. Sometimes we have cases of missing dog pets which could have been easily found if only we had a trained recognition system that can identify and alert humans if such a dog is seen on camera. This is the problem I am trying to solve with CNN.

Datasets

I will use the dataset from these sources:

The dog dataset . The human dataset .

Solution Statement

The goal of this project is to create a dog breed identification application using convolutional neural networks. The resulting algorithm could be used as part of a mobile or web app. This design can be applied to image classification problem.

The code will accept any user-supplied image as input. If a dog is detected in the image, it will provide an estimate of the dog's breed. If a human is detected, it will provide an estimate of the dog breed that is most resembling.

Benchmark Model

I will use AlexNet and ResNet50 transfer learning models a benchmarks.

Evaluation Metrics

The evaluation metrics of this problem will be the Accuracy Score for the designed algorithm and the model that give us the lowest validation loss.

Project Design

The overall flow consists of the following steps:

- **Import Datasets:** Our first step will be to load the datasets that are divided into train, validation and test folders.
- **Detect Humans:** In this section, we use OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces in images.
- **Detect Dogs:** In this section, we use a VGG-16 model to detect dogs in images.
- **Create and train a CNN to classify Dog Breeds**: In this step, we will create a CNN-based model from scratch that classifies dog breeds. In the next step, we will use an approach based on transfer learning.
- **Train a CNN to Classify Dog Breeds (via transfer learning):** We will now use transfer learning to create a CNN that can identify dog breed from images. We will reuse the data loaders that we created earlier.
- **Dog breed classification algorithm:**
    - 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 in the image, provide output as other thing.
- **Testing the algorithm:** Lastly, We will test our algorithm on some sample images to measure the performance of our algorithm.