Machine Learning Engineer Nanodegree
Capstone Proposal

Kiran Vasishta TS
April 6th, 2020

Dog Breed Classifier

Domain Background

Dog breed classifier is one of the popular problems in the Convolutional Neural Networks. The main idea behind this problem is to clearly explore and use the capabilities of the Deep Learning neural networks to get the best classifier to predict the breed of the dog using its images. So, I selected this problem domain since I really want to explore this area of Machine Learning and further use the skills, I obtain here for the purpose of solving other real world challenges like identification of the crop diseases using the image of the diseased plant.

Problem Statement

The Dog Breed classifier problem is the scenario where the dog-images are fed to the app and we must classify as to which breed the dog in the given image belongs to. Now the real challenge is that, the image may not be an image, it can also be a human. Our model must be able to correctly identify this as well. If the image is human, the model must determine as to which breed of the dog the human resembles.

Datasets and Inputs

The datasets are provided by the Udacity and Kaggle. We can take either of those. For this I am using the data provided by Udacity for the same. Please refer to the below links for downloading the data. Also, we need the human data to train the model to detect the human faces.

Dog Data:
https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/dogImages.zip

Human Data:
https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/lfw.zip

Here I will provide the sample data for the same. This is just the sample data from the large dataset of approximately 1GB of data.

Input Size: All the images are of the size 250X250. There is a total of 133 breeds.

Solution Statement

To explain my solution statement, I must first explain the nature of the problem. The problem is the multiclass one since there are a total of 133 breeds thus 133 different classes. So, we will use the Convolution Neural Network (CNN). The CNN is the algorithm which accepts the image, differentiates between various features (like car in the image where the dog is sitting next to car) and detects the main feature of the image.

In this application I am planning to use a pretrained VG16 model to detect the dog images. Also, for the other part of our dataset, the human images we use the existing algorithms like OpenCV's implementation.
Once this is done, we can use the model thus created to predict the breed of the dog or if human, then detect the breed he/she resembles. Also, it is neither, then output an appropriate message.

Benchmark Model

In this application we will follow 2 steps.

- In the first step we will create our own model which has to have at least 10% accuracy. This conforms that the model is working as expected since the random guesses will lead to successful prediction in 1 out of 133 runs since there are 133 breeds.
- In the second step we will use the transfer learning method and try to achieve the accuracy of up to 60%. The Architecture used here is Resnet.

Evaluation Metrics

For this use-case I believe that the accuracy cannot be trusted for the measurement of the performance since the dataset used is imbalanced. So, we must use other metrics like log-loss function to measure the performance. This is a good choice in my opinion since the log loss function considers, as to how much the variation is present when compared with the actual label. This is particularly helpful in the situation where the dog breeds are similar, like different types of Retriever breed look similar.

Project Design

For this project I have planned to follow the below steps:

- As with all the data-science and machine learning project this project too begins with the data preprocessing.
- The data will be split into the Train, Test and Validate data.
- Then comes the main process, the image augmentation. This is done to get the variations of the images. This helps to predict the cases where the dogs in the images are in different angles to the frame.
- Detect the human faces using the OpenCV implementation of the classifiers.
- VG16 model is used to create the dog detector.
- The CNN is model is created from scratch. This includes the training, testing and validating the model.
- A CNN model with Resnet101 architecture is used to create the model using the transfer learning process.
- Finally, the main code of this app is written. Here the dog detector and human detector is combined, and the following cases are handled.
    1. If the dog is detected, then output the breed.
    2. If the human is detected, then output the breed the human resembles, along with the message that the human is detected.
    3. If neither, then output appropriate message indicating the same.

Reference

1. Udacity dog-breed-classifier https://github.com/udacity/deep-learning-v2-pytorch/tree/master/project-dog-classification
2. Resnet101 from Kaggle: https://www.kaggle.com/pytorch/resnet101
3. Pytorch tutorials: https://pytorch.org/