Capstone Project Hemang Goyal

Machine Learning Engineer Nanodegree 25/5/2020

Dog Breed Classifier using CNN

Domain Background

The Dog breed classifier is a well-known problem in Machine Learning domain. Given an image of a dog, the classifier has to identify an estimate of the canine's breed. If supplied an image of a human, the classifier will identify the resembling dog breed. The goal is to build a pipeline which can process real world user supplied images and identify an estimate of the canine's breed. We can use supervised machine learning algorithms to solve this problem as this is a multi-class classification problem. After completing the training and testing of the model, I am planning to deploy this model through as web app where a user can provide an image as an input and obtain prediction from this model. I have chosen this as my capstone project since it is giving me an opportunity to build as well as deploy my Machine Learning Model to common users.

Problem Statement

The goal of this project is to build a machine learning model which will act as a classifier to classify images which can be used within web app to process real-world, user-supplied images.

The algorithm has to perform two tasks:

Dog face Detection: Given an image of a dog, the algorithm will have to detect that it is a dog's face and predict an estimate of the dog's breed.

Human face Detection: If supplied an image of a human, the algorithm will identify that it is a human face and then will try to find the dog breed from the model which will resemble the most or closely to that human face.

Datasets and Inputs

For this project, the input format must be of an 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 itself. It contains pictures of dogs and humans.

Here are the details about the dataset :

Dog images dataset*:*

Total images : 8351

Train images : 6680

Valid images : 835

Test images : 836

Each of this directory (train, test, valid) have 133 folders corresponding to dog breeds. The images are of different sizes and different backgrounds, some images are not full-sized. The data is not balanced because the number of images provided for each breed varies. Few have 4 images while some have 8 images.

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. The data is not balanced because we have 1 image for some people and many images for some.

Solution Statement

We can use Convolutional Neural Network to solve this problem and perform multi-class classification. A **Convolutional Neural Network (CNN)** is a Deep Learning algorithm which can take in an input image, can learn various features in the image (using learnable weights and biases) and using this features, it can be able to differentiate between different images and can help classify them.

The solution involves three steps :

1) Detect human images : To do this, we can use existing algorithm like OpenCV's implementation of Haar feature based cascade classifiers.

2) Detect dog-images : We can do this using a pretrained VGG16 model.

3) After the image is identified as dog/human, we can pass this image to an CNN which will process the image and predict the breed that matches the best out of 133 breeds.

Benchmark Model

- ✓ The CNN model created from scratch must have accuracy of at least 10%. This can confirm that the model is working because a random guess will provide a correct answer roughly 1 in 133 times, which corresponds to an accuracy of less than 1%.
- ✓ The CNN model created using transfer learning must have accuracy of 60% and above.

Evaluation metrics

Multi class log loss function will be used to evaluate this multi-class classification model. Accuracy is a not a good indicator here to measure the performance because of the imbalance in the dataset. Since Log loss takes into account the uncertainty of prediction based on how much it varies from actual label, it 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. Perform Image augmentation on training data.

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

Step 5: Create a CNN to Classify Dog Breeds using Transfer Learning with resnet101 architecture. Train and test the model.

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

References

1. Original repo for Project - GitHub*:*

   https://github.com/udacity/deep-learning-v2- pytorch/blob/master/project-dog-classification/

2. About Resnet101 architecture*:*

   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. CNN:

   https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural- networks-the-eli5-way-3bd2b1164a53

5. About Log loss function *: http://wiki.fast.ai/index.php/Log_Loss*