Capstone Project
Machine Learning Engineer

Suganthan Arulvelan
19th October 2020

Dog Breed Classifier Using CNN

Description

The Dog Breed Classifier is one of the best Machine Learning Projects. This project is about finding the dog breed based on image of the dog. If a human image is given, it finds the close match of the dog breed and outputs the name of the breed. It is a multi-classification problem,Here we are using supervised machine learning model to solve this problem.

The aim of the model is to build an algorithm that performs two tasks.

Dog face detection: Provided with a dog image to the model, the model will identify and estimate the canine's breed.

Human face detection: If a human image is provided the model will identify the resembling dog's breed.

Datasets and Inputs: 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 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

To perform this multi-class classification, Convolution Neural Network(CNN) is used to solve the problem. CNN is a deep learning algorithm that takes in the input image and assign importance to various aspects in the image which is able to differentiate one and another.

Human Image Detection: To detect the human image we are using an algorithm Called OpenCV's implementation of haar feature based cascade classifiers.

Dog Image Detection: To detect the dog image pre-trained VGG16 model is used.

After identifying the image between dog and human the image is passed to CNN which will process it and predict the breed that is well matched out of 133 breeds.

Benchmark Model : The CNN model from scratch should have an 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.

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 pre-trained 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.

- If dog is detected in the image, return the predicted breed.
- If human is detected in the image, return the resembling dog breed.
- If neither is detected, provide output that indicates the error.

References

1. Original repo for Project - GitHub: https://github.com/udacity/deep-learning-v2- pytorch/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/97304e232807082c2e7b54c597615dc0ad8f 6173/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