Udacity Machine Learning Engineer

Dog Breed Classifier

October 30, 2020

Domain Background

Artificial Intelligence has been witnessing a monumental growth in bridging the gap between the capabilities of humans and machines. The advancements in Computer Vision with Deep Learning has been constructed and perfected with time, primarily over one particular direction a Convolutional Neural Network. A CNN is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

Problem Statement

Biodiversity is one of the most complex and vital features of our planet. Biodiversity is a modern term which simply means " the variety of life on earth". This variety can be measured on several different levels. Among these diverse animals we call one a man's best friend. Even though they are all individually different we can differentiate them from their visual and other features into different breeds. We wish to train a machine to be able to recognise and classify these numerous species. Also for fun, we will try to classify humans into dog breeds that the trained model thinks looks similar to them.

Datasets and Inputs

The dataset is contained within the workspace provided by Udacity machine learning engineer nanodegree program.

The provided dataset comes with various dog pictures along with their breed labels as well as numerous human pictures.

Further transfer learning will be used to first differentiate between dog and human and then between the breeds. Two pre-trained models VGG-16 (PyTorch) and ResNet50 will be used for the purpose.

The dog breed dataset includes the following components:

1. Train - The training dataset contains 133 dog breeds with 30-70 pictures for each breed.

2. Test - The test dataset contains the same 133 dog breeds with 6-10 pictures for each breed.

3. Validation - The validation dataset contains the same 133 dog breeds with 6-10 pictures.

The human dataset contains 5749 different pictures of humans.

The dataset not only contains isolated images of dog and human in their respective datasets but also images that have both dog and human and various other animals too.

Solution Statement

The solution of this project will include a trained detector using CNN to detect dog breeds based on the supplied picture. The accuracy rate is expected to be above 90%. The model is expected to tell if the dog is pure breed or mixed breed based on feature scores.

The model is expected to tell humans apart from dogs. As often used by the Kaggle competition, the model will be evaluated by the cross entropy loss function.

Benchmark

As per the provided guide Jupyter notebook that the transfer learning model should have an accuracy greater than 90%.

Similarly the neural network designed by hand should have an accuracy greater than 10% to pass the assignedcr task.

Project Design

The project will follow the steps provided by the standard project template:

Step 0: Import Datasets.

Step 1: Develop components using Haar Cascade classifiers to detect humans.

Step 2: Develop components using a pre-trained VGG-16 model to detect dog breeds.

Step 3: Assemble components to a full detection algorithm.Step 4: Test the completed algorithm.

The total dataset (include training, test, and validation) contains 133 breeds of dog with

approximately ~8.3k of images.

Before constructing the CNN model for dog breed classification, the dataset is explored by testing the pre-trained model, i.e., cascade classifier and VGG-16 to detect whether. The image contains humans or dogs. The images will be passed through the human and dog detectors before passing into the breed classifier. If no dogs are detected in the image, the image will still be passed into the breed classifier and the resultant most probable breed will be returned for fun context .

References

https://docs.opencv.org/3.4/d7/d8b/tutorial_py_face_detection.html

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

https://neurohive.io/en/popular-networks/vgg16/

https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035