Capstone Project

Jawaher Madkhali

Machine Learning Engineer Nanodegree

January 23, 2022

Dog Breed Classifier using CNN

Domain Background

One of the problems faced by those interested in this field is determining the breed of dogs, which is a well-known problem in machine learning. The problem lies when the input images of dogs, and human images are provided, it is important to identify the breed of dogs that resemble them, this is done by building a CNN that can process images Provided by the user from the real world and dog breed prediction, there are certainly many challenges in this model that would be interesting to work on, and try to delve into this topic to understand the work in the real world.

Problem Statement

The main goal is to build a model to process images that the user provides from the real world, it requires two main tasks:

Human Face Detector: If human images are provided to predict them, the breed of dogs they resemble will be identified.

Dog face detector: The main goal is to detect the dog's breed, after identifying the images of the dog and the human, these images must be processed and the process of predicting the type of breed.

Dataset and Input

The input must be of the type of images because we want to do the prediction and determine the type of dog breed, the data for this project was provided by Udacity, a large dataset containing a number of dog and human images was provided:

- Human image data: The human image data set contains 13,233 total images arranged according to human names, The images differ from each other in angles, background, number of people, and all images have a size of 250*250.

-Dog image data: The dog data set contains 8351 total images sorted in the train 6680 images, the test 836 images, valid 835 images, all in a folder containing 133 correspond to dog breeds, the images differ in sizes and backgrounds and some images have more than one breed.

Solution Statement

To solve the problem, we must complete the three main sections of our project:

Detect humans and detect dogs and detect dog breeds

1- To detect human images we use OpenCV Cascading Classifiers based on Haar features to detect human faces in images. OpenCV provides several pre-trained face detectors, which are stored as XML files on Github.

2- To detect dogs we use a pre-trained model, the VGG-16 model, which has been trained on the ImageNet dataset, a very large and very common dataset used for image classification and other vision tasks.

3- Finally, we reach the main goal, which is to detect the dog's breed, after selecting the dog and human images, we pass these images to the CNN model, which will process these images and predict the breed that matches the best among 133 breeds.

Benchmark Model

The required conditions serve as criteria in our model: The first condition is that the convolutional neural network made from scratch needs an accuracy of at least 10%.

The second condition is that the pre-trained model has an accuracy of at least 60% so that it can be used successfully in a dog breed classifier.

Evaluation Metrics

For this classification, accuracy is not a good indicator for measuring performance, because accuracy may differ from one person to another depending on the tools used and experience in this field.

Project Design

Step 1: Import Datasets

Step 2: Detect Humans we use OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces in images.

Step 3: To Detect Dogs we use a pre-trained model, the VGG-16 model

Step 4: Create a CNN to Classify Dog Breeds (from Scratch)

Step 5: Create a CNN to Classify Dog Breeds (using Transfer Learning)

Step 6: Write an algorithm that accepts a file path to an image and first determines whether the image contains a human, dog, or neither. Then,

-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 that indicates an error.

References

1- https://github.com/udacity/machine-learning/blob/master/projects/capstone/capstone_proposal_template.md

2- https://pytorch.org/docs/stable/data.html#torch.utils.data.DataLoader

3- https://pytorch.org/docs/master/nn.html#loss-functions