Capstone Project

Machine Learning Engineer Nanodegree

Lucas Micol

October 5th, 2020

Dog Breed Classifier

Project Overview

In this project, we will combine models of human and dog recognition in a dog breed classification algorithm, capable of detecting if an image contains a dog and providing its breed, or detect a human and estimate the dog breed that is most resembling.

Problem Statement

The goal of this project is to combine methods of image recognition to solve the classification of dog breeds. The idea is to identify if a photo contains a dog or a human and estimate the breed of dog or the dog breed that most resembles the person.

For this objective, multiple machine learning techniques will be combined into one final algorithm, including Convolutional neural networks[1], OpenCV[2], VGG-16 Model[3] and PyTorch transferred learning models[4]. The expected end result is an algorithm that has an accuracy above 60% and that can be later deployed on an AWS Endpoint.

Data Exploration

To solve this problem we have two datasets, one with photos of humans and other with dog pictures. The formers contain 13233 images that are organized in 5750 folders with the name of that person. The latter contains 8351 images distributed in 133 folders corresponding to the dog breeds.

For both cases the data is not balanced. However, the images are standardized and have the size of 250x250 pixels. Below we present an example in Figure 1 and Figure 2 of those datasets.

[1] https://cs231n.github.io/convolutional-networks/

[2] https://opencv.org

[3] https://neurohive.io/en/popular-networks/vgg16/

[4] https://pytorch.org/docs/stable/torchvision/models.html

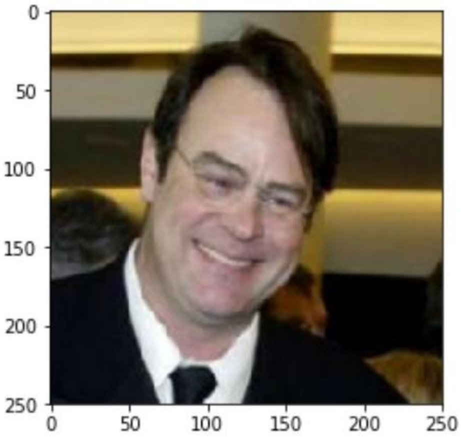

Figure 1: Human photo example

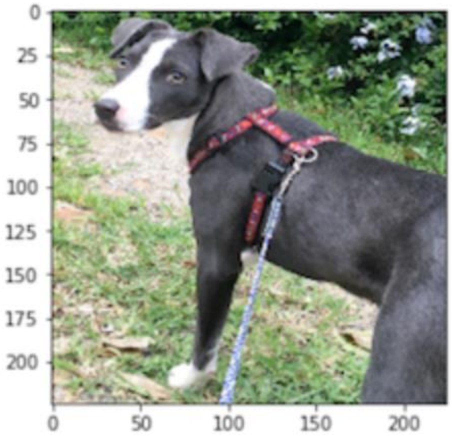

Figure 2: Dog photo example

Evaluation and Benchmark

For this project there will be two implementations. The first being a CNN made from scratch that must reach at least **10%** of accuracy. And the second one being a CNN using transfer learning that need to achieve **60%** or above.

$$Accuracy = \frac{(true\ positives + true\ negatives)}{dataset\ size}$$

In the evaluation part, considering that we have multiple categories, the indicated loss function is the **cross-entropy loss**, also known as log loss.

Methodology

We have seven defined steps to achieve the goal of this project. Being these:

1. Import the libraries and explore the datasets. This part also includes data pre-processing, such as the separation between training, testing and validation
2. Detect humans faces using OpenCV's implementation
3. Detect dogs using pre-trained VGG-16 Model
4. Create a CNN from scratch to classify dog breeds
5. Use transfer learning to create a CNN to classify dog breeds
6. Write the final algorithm that combines the detection of dogs and humans (If none is recognized, the algorithm must indicate an error).
7. Test the algorithm and validate the results

Improvement

With the completion of this work, I would like to test my learning by deploying the developed model on an endpoint on AWS. And together with that, host a website on GitHub that communicates with a REST API and a Lambda function on Amazon server to apply an image sent to the neural network and return the result to the user in a web application. However, due to the schedule of the nanodegree, this idea remains as future work.

References

1. Source code of the project: https://github.com/udacity/dog-project
2. PyTorch documentation: https://pytorch.org/docs/stable/index.html