Alain Víctor Ramirez Martínez
February 22, 2021

DOG BREED IDENTIFICATION WITH CNN

Domain Background

From the moment we open our eyes, we gradually learn how to classify all the objects around our living space. For some people, this ability is unique to humans and most of the living beings. ¿What if I told you that Machine Learning allows machines to do the same classification with an accuracy even superior to human eyes? The arise of modern CNN (Convolutional neural network) models in the recent years had allows us to apply this technology in a variety of fields. In this project, I will address the task of dog breed identification. My motivation is to demonstrate how powerful is CNN for a multi-class classification problem.

Problem Statement

The main goal of this project is to deploy a model than can classify a dog breed from an image. The first step is to distinguish whether the image contains a human face or a dog. After that, the next step is to detect the dog breed with at least 60% of accuracy. Our final solution must distinguish a dog breed when a dog is present in the image or suggest the most similar dog breed when the image contains a human face. The final part is done just for fun.

Datasets and Inputs

The dataset for this project is provided by Udacity machine learning nanodegree program. The information is shared in a repository. It contains the following folders and files:

- /dog images with subfolders: train, valid and test with 8351 total dog pictures.
- /lfs with 13233 total human pictures.

The dog's folder is the most important for our purposes. Each subfolder has 133 subfolders which represent the 133 corresponding dog breeds. It is important to note that all the images in these folders are in different sizes and angles.

Solution statement

The first part of our solution is done with the help of the OpenCV library and the Haar Cascades feature. Its face detector has a performance of almost 100% if a human face is provided. The second part for dog detection requires to load the VGG16 pre-trained model. The performance of this model is almost 100% if a dog image is provided. The last part which addresses our problem, the dog breed classifier, is done deploying a model from scratch and then using transfer learning to improve that original model. The model from scratch was obtained from Sushant Agarwal. I choose Alexnet from Alex Krizhevsky as a model for transfer learning. Finally, the algorithm designed for our solution uses the dog and face detectors to filter the image provided and then predict the dog breed.

Benchmark Model

Due to the model from scratch required at least 10% of accuracy, I decided to use any 2-dimensional CNN (Conv2d) designed for image classification. It is a good accuracy compared to the one obtained from a random guess in 133 dog breeds (less than 1%).

For the transfer learning model, I used a benchmark (see references section) and chose the one with the highest Top-1 error (single-crop error rate on the ILSVRC 2012 validation set) to demonstrate the power of transfer learning technique.

Evaluation Metrics

Our main goal is to detect a dog breed from a given image or a set of images, so “accuracy” is going to be our metric. Specifically, for the two CNN models deployed:

- Test accuracy greater than 10% for the model from scratch.
- Test accuracy greater than 60% for the transfer model.

Project Design

This project follows the steps provided by Udacity nanodegree program:

Step 0: Import Datasets:

The folders, subfolders and files are imported and counted.

Step 1: Detect Humans:

By using the face detector feature from OpenCV library and HaarCascade classifier.

Step 2: Detect Dogs

By using the pre-trained VGG-16 model on ImageNET.

Step 3: Create a CNN to Classify Dog Breeds (from Scratch)

By using the pytorch framework and pre-processing the dataset provided (resize, crop, flip). The normalization is configured as part of the transforms.

The batch size was set to 20 because the low number of samples.

The model has 3,296,381 trainable parameters.

This model predicts dog breed with 16% of accuracy in the Test Data.

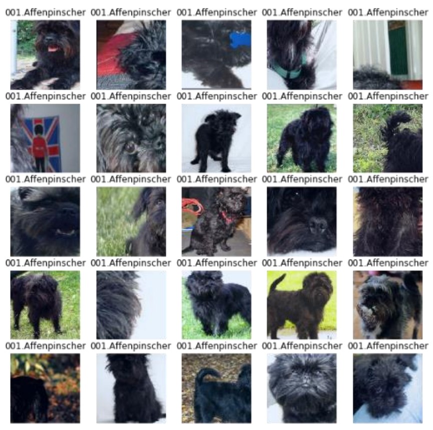

Fig1. Test Data after transforms.

Step 4: Create a CNN to Classify Dog Breeds (using Transfer Learning)

By using the pre-trained AlexNet model as a basis and the loaders of our trained model. The output features originally is set to 1000, it is necessary to change to 133.

Step 5: Write Your Algorithm

By using the transfer learning mode and the dog and face detectors integrated in a simple algorithm that first load an image from a folder and then distinguish if it contains a dog or a human face. Then predict the dog breed.

- 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.

Step 6: Test Your Algorithm

By using the algorithm for Step 5 in a set of personal images.

Result:

```
Great news! Dog detected!
I think it is a Miniature schnauzer
```

References

- Capstone Project Example: https://github.com/udacity/machine-learning/blob/master/projects/capstone/report-example-3.pdf (Shibuya,2020).
- CNN-Benchmarks: https://github.com/jcjohnson/cnn-benchmarks (Johnson, 2017).
- Model from Scratch reference: https://medium.com/analytics-vidhya/classification-of-dog-breed-using-deep-learning-343f98ebebf0 (Agarwal, 2020).