Machine Learning Engineer Nanodegree

Dog Breed Classifier Capstone Project Report

Raghad Hethnawi

October 19th, 2020

1. Definition

1.1 Project Overview

Dog breed classification is one of the well-known problems in the field of Machine Learning. Its importance arises from lack of awareness that some people have due to their dogs' breeds, where dog breed should be identified correctly while considering buying dog's food and other stuff. Therefore, each dog owner should at least know the breed(s) of his dog(s).
This can be done by image classification, which is the process in deep learning where it tries to classify images due to its visual contents. Convolutional Neural Network will be used for such a task.

1.2 Problem Statement

Image classification is the process that starts from taking an input, applying some processes on it, and outputting it as a class. For example, taking an image as input, applying CNN to it, and output the result as a class of a dog. It's easy to us as humans to know that a specific picture represents something (a cat or a dog for example), but how can machines do that?

1.3 Metrics

One of the main metrics used to measure the performance of the model is the model's accuracy. It can be calculated as follows:
Accuracy = (Number of correctly classified images/Total Number of images) * 100%

But, since the given data is not quite balanced, the evaluation model is based upon multi-class log loss, where the log loss is going to take into account the uncertainty of the prediction based on how much it varies from the actual label.

2. Analysis

2.1 Data Exploration

The data included for this project has image type, since we want to determine the dog's breed detected in that image(if the image was for a dog), or the resembling dog breed for the human in that image (if the image was for a human).

Two main datasets were included, human images dataset, and dogs images dataset.

- **Human Images Dataset:** Which includes 13233 total of human images. These images are sorted by names of humans, and spreaded in 5750 folders. Images have different angles and different backgrounds, but they're all of size 250×250. Worth mentioning is that the data is unbalanced, since some people have only one image, while others have more than one.

- **Dog Images Dataset:** Which includes 8351 total of dog images, sorted in train(6680 images), valid(835 images), and test(836 images) directories. Each one of these directories include 133 folders corresponding to dog breeds. The number of images included for each breed varies, hence, the data is not balanced. Images also have different sizes and different backgrounds.

2.2 Sample images from the dataset

Figure 1: Sample dog image

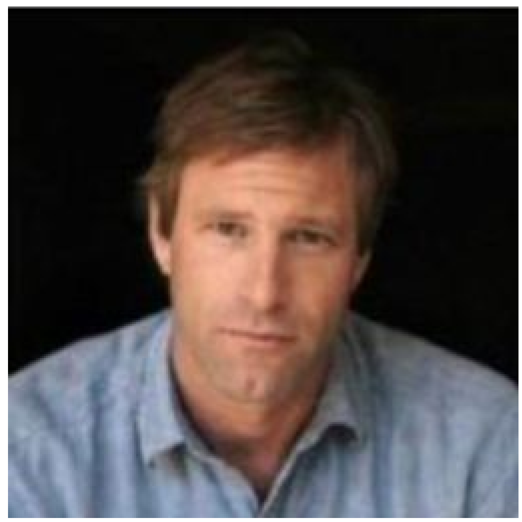


Figure 2: Sample Human Image

3. Algorithms and Techniques

3.1 Convolutional Neural Network (CNN)

A convolutional Neural Network is a common deep learning algorithm that we've used for this multiclass classification.

CNN takes inputs as a form of images, assigns learnable weights for the various objects/aspects of the image, in order to differentiate one from another.

3.2 Methodology

3.2.1 Import Datasets

- Downloading the dog dataset, unzip the folder in order to use it in the home directory, located at /dog_images
- Downloading the human dataset, unzip the folder in order to use it in the home directory

3.2.2 Detecting Human Images

- OpenCV's implementation of Haar feature-based cascade classifier is used in order to detect human face images - there are many pretrained face detectors stored as XML files on GitHub
- one of those detectors is downloaded and stored in the haarcascades directory, in order to demonstrate how to use this detector, to detect human faces in images.
- Building human detector function using that classifier

3.2.3 Detecting Dog Images

- Used the pre-trained model mentioned earlier to detect dog's breed in this case VGG 16 model.
- Writing the code to make predictions whenever an image is given
- Building the dog detector and assessing it.

3.2.4 Implementing CNN model to classify dog breeds(from scratch)

- Specified three separate data loaders for the training, validation, and test datasets of dog images (located at dog_images/train, dog_images/valid, and dog_images/test).
- The project input is a RGB image which cropped with 224x224 pixel and depth (3) (the 3 colors), the project input has the shape of 3x224x224
- Building the model architecture, specified the loss function and optimizer.
- Used the flatten image input view function in order to reshape the tensor.
- Three fully connected Linear layers with ReLU activation function were added, and dropout layers for the hidden layers with a percentage of **50%** to avoid the bias.
- Training and validating the model
- Testing the model

3.2.5 Creating a CNN model to Classify Dog Breeds (using Transfer Learning)

- Following the same steps in 3.2.4
- Downloading the VGG16 and modified the last fully connected layers to suit our goal
- Implementing the model, and predicting the dog breed using the model.

3.2.6 Writing my algorithm

Crating the model that accepts an image as an input, detect whether it's a human image, a dog image, or anything else, after that:

- If the image is a dog image, the model returns the predicted breed of that dog.
- If the image is a human image, the model returns resembling the dog breed of that human image.
- Otherwise, if the image wasn't either a dog image, nor a human image, the model returns that it didn't recognize that image.

3.2.7 Testing the model with new images
In this step, the model is tested by providing new images as input, with 2 different types (human images and dog images) to test its performance due to the predicted breed.

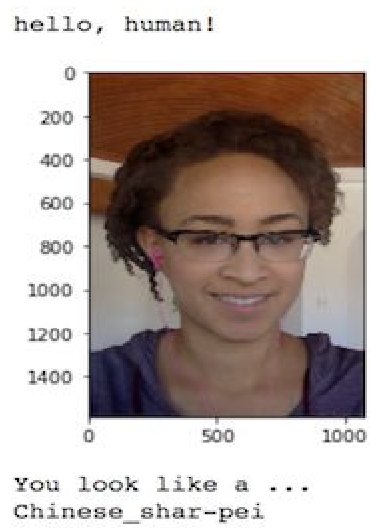


Figure 3 : Sample Dog Output   Figure 4 : Sample Human Output

4. Benchmark

1. The CNN which is created from scratch must have at least accuracy of 10%.
This informs that the model is working properly, where a random guess gives a correct answer roughly once in 133 times, giving a very low accuracy (less than 1%).

2. The CNN model which is created by the Transfer Learning should have accuracy of 60% or above.

5. Results

The CNN model created to classify dog breeds (from scratch) surpass the threshold of 10%, which had:

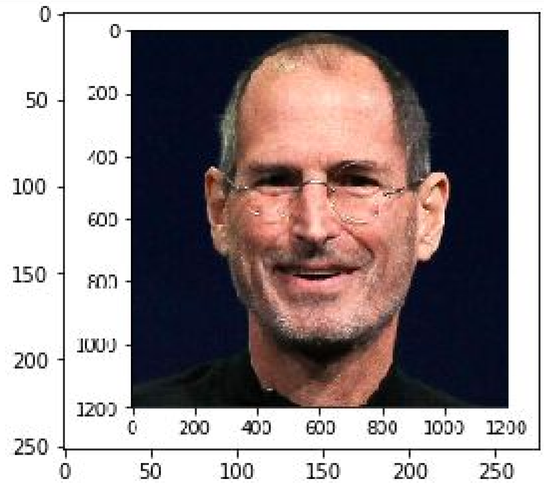
- Test Loss: 3.668755
- Test Accuracy: 14% (123/836)

Whereas the CNN model to Classify Dog Breeds (using Transfer Learning) surpass the threshold of 60%, which had:

- Test Loss: 0.616596
- Test Accuracy: 82% (693/836)

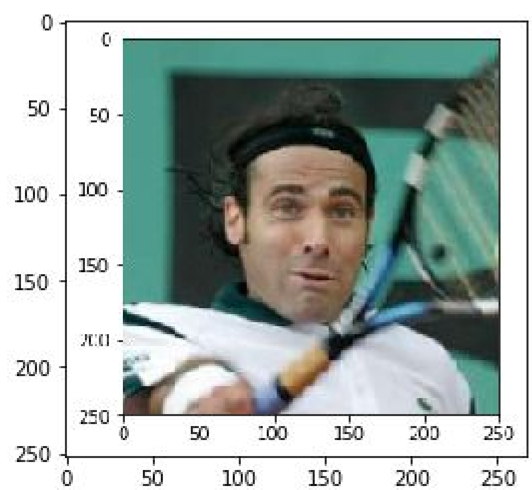
6. Samples Testing the Model

The following samples shows the results of testing the model with new images (for both human and dogs images)



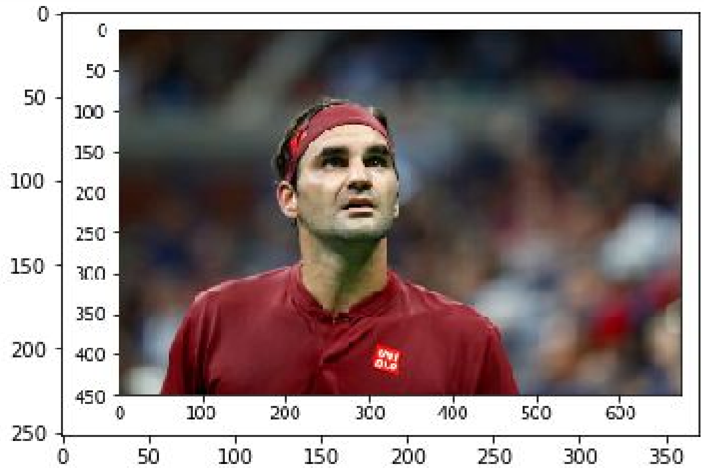
```
Hello, human!
If you were a dog..You would look like a Bullmastiff
```

Figure 5 : Model Testing Result of Human image



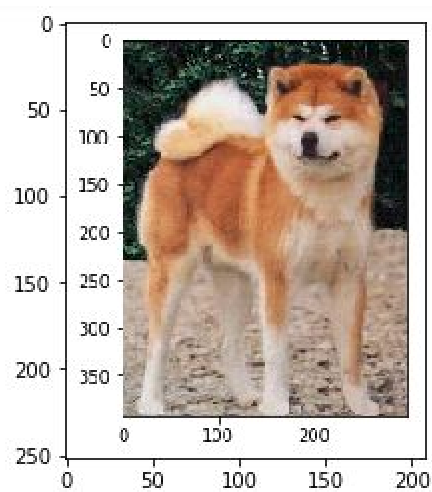
```
Hello, human!
If you were a dog..You would look like a Bullmastiff
```

Figure 6 : Model Testing Result of Human image




```
Hello, human!
If you were a dog..You would look like a Bullmastiff
```

Figure 7 : Model Testing Result of Human image




```
A Dog is Detected!
It looks like a Akita
```

Figure 8 : Model Testing Result of Dog image

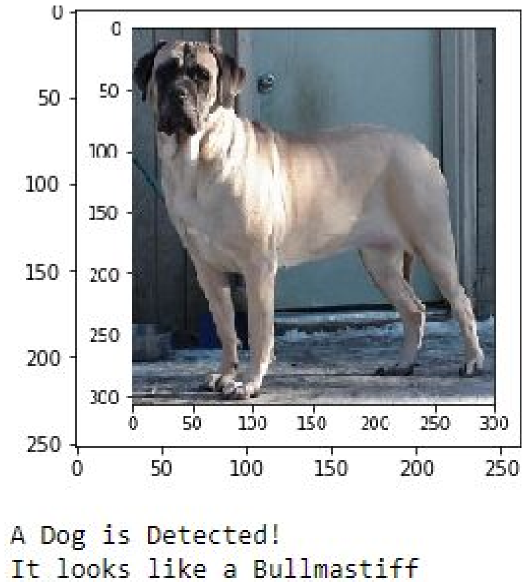


Figure 9 : Model Testing Result of Dog image

7. Bibliography

1. https://glassboxmedicine.com/2019/04/13/a-short-history-of-convolutional-neural-networks/
2. https://pytorch.org/docs/master/
3. https://www.kaggle.com/gpreda/haar-cascades-for-face-detection
4. https://arxiv.org/abs/1409.1556
5. https://arxiv.org/abs/1409.4842
6. https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf