Author: Md Imranur Rahman Date: 24-July-2021

Udacity Nanodegree

Machine Learning Engineer

Report

CNN Project: Dog Breed Classifier

Definition

Project Overview

In this capstone project I will build a machine learning model which will process images. If an image of a dog is provided, the model will identify its breed. If an image of a human is provided, the model will identify the most resembling dog breed.

Problem Statement

Images of dog and humans will be provide as an input to the machine learning model.

1. **Dog Image:** If an image of dog is given, then the algorithm will identify it as a dog. It will then detect the breed of that dog.

2. **Human Image:** If an image of human is given, then the algorithm will identify it as human. After that the model will match it with the most resembling dog breed.

Metrics

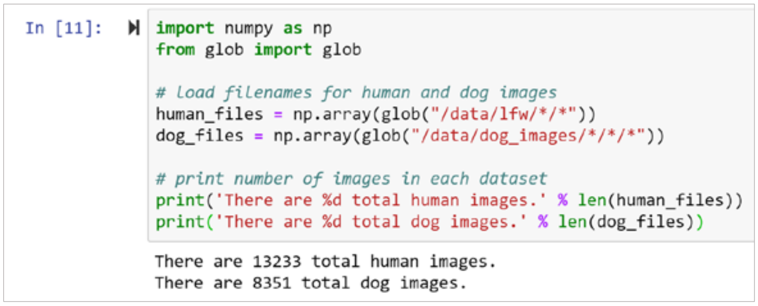
Here I’ve used accuracy as a metric to evaluate the performance of the model.

Accuracy = Correctly identified items / All Items

On another note, during the model training validation loss was calculated. And model file was saved when validation loss decreased.

Analysis

Data Exploration AND Exploratory Visualization



In [11]:

```
import numpy as np
from glob import glob

# load filenames for human and dog images
human_files = np.array(glob("/data/lfw/*/*"))
dog_files = np.array(glob("/data/dog_images/*/*/*"))

# print number of images in each dataset
print('There are %d total human images.' % len(human_files))
print('There are %d total dog images.' % len(dog_files))
```

```
There are 13233 total human images.
There are 8351 total dog images.
```

2 datasets were used. These are provided by Udacity.

1. **Human Dataset:**

   - Images are of size 250x250 pixels
   - Images have different backgrounds
   - Dataset is not balanced (as 13233 images of 5750 persons)

   Sharing the relevant numbers below:

   a. Total Image: 13233
   b. Unique Person: 5750

   Sample Image:

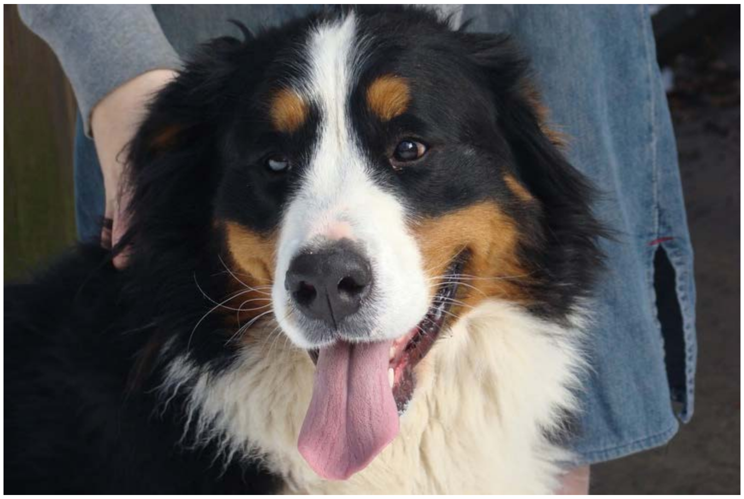
2. **Dog Dataset:**

- Images are of different sizes
- Images have different backgrounds
- The dataset is not balanced as-well (sample images of each breed are not equal)

Sharing the relevant numbers below.

a. Total Image: 8351
b. Dog Breeds: 133
c. Training Set: 6680
d. Test Set: 836
e. Validation Set: 835

Sample Images:



Algorithms and Techniques

A Convolutional Neural Network (CNN) model has been built as a solution. This CNN model estimated the breed of the provided dog image. If image of human is provided, instead of a dog image, then it'll mark it as human and will find the most resemblance with a dog breed.

To mention the technique part: Open CV's implementation has been used to identify faces. VGG16 model was used to identify dogs. CNN model identified the Dog Breed.

Benchmark

1. **CNN Model created from scratch:** It must have an accuracy of 10%.

2. **CNN Model created using Transfer Learning:** It must have an accuracy of 60%.

Methodology

Data Preprocessing

1. In the code I've resized the images to 256x256 pixels.
2. Then random cropping was done to 224x224 pixels, since VGG-16 takes input in this size.
3. I have decided to augment the data set:

I. I've used 20 degree rotation

II. Then horizontal flip to reduce overfitting.

Implementation

I built a CNN model from scratch to implement the solution approach.

- 4 convolutional layers has been used
- Size of convolving kernel is 3
- Stride of the convolution is 1 (default)
- Number of channels in the input image are 3, 36, 64, 128 respectively for each layer
- Number of channels produced by the convolution are 36, 64, 128 respectively for each layer
- Padding added to all four sides of the input = 1
- The model produces 133 dimensional outputs
- 20% dropout has been considered

This model gave an accuracy of 12% after 10 epochs.

Refinement

To increase the accuracy I used the transfer learning model. ResNet-101 architecture was selected for this. It comes with 101 layers. I increased the out_features to accommodate 133 layers. After 5 epochs the accuracy came as 82%, which is very good compared to the benchmark model. Increasing epochs would increase the accuracy I believe. This architecture is suitable because rsnet101 is already pre-trained.

Results

Model Evaluation and Validation

Human Face Detector:

Classifier: haarcascades/haarcascade_frontalface_alt.xml

```
Percentage of the first 100 images in human_files have a detected human face:  98
Percentage of the first 100 images in dog_files have a detected human face:  17
```

Classifier: haarcascades/haarcascade_frontalface_alt2.xml

```
Percentage of the first 100 images in human_files have a detected human face:  100
Percentage of the first 100 images in dog_files have a detected human face:  21
```

Classifier: haarcascades/haarcascade_frontalface_alt_tree.xml

```
Percentage of the first 100 images in human_files have a detected human face:  57
Percentage of the first 100 images in dog_files have a detected human face:  2
```

Considering the result I used the first classifier, as it provided less incorrect data in dog_files.

Dog Face Detector:

Pre-train VGG16 model has been used which achieved below result.

```
Percentage of the images in human_files_short have a detected dog:  2
Percentage of the images in dog_files_short have a detected dog:  100
```

CNN from Scratch:

```
Test Loss: 3.765670
Test Accuracy: 12% (106/836)
```

CNN using Transfer Learning:

```
Epochs: 5
Test Loss: 0.690242
Test Accuracy: 82% (689/836)
```

Justification

An accuracy of 82% seems very good to me. The output is better than I expected. It can classify human and dogs. After providing an image of a cat and a cycle it showed the error message.