Capstone Project

Jawaher Madkhali

Machine Learning Engineer Nanodegree

January 23, 2022

Dog Breed Classifier using CNN

I. Definition

Project Overview

Dog Breed classifier project of the Machine Learning Nanodegree by Udacity. in which the user can check whether the uploaded image is a dog or human image, the deep learning model distinguishes between 133 dog classes using convolutional neural networks side by side to achieve results.

Problem Statement

The problem is identifying the breed of dogs, if a dog image is provided as input, and if a human image is provided, then we have to identify a similar dog breed. The idea is to build a pipeline that can process images that the user provides in the real world and determine an estimate of the breed of dog.

Metrics

The metric that we use to measure how well our model is performing in classifying the dog breed of images is Accuracy. Accuracy is the items classified correctly by our model. It can be percentage of total given as

$$accuracy\ \% = \frac{total\ number\ of\ correct\ predictions}{total\ number\ of\ images}$$

Accuracy would be an appropriate metric for our model, as it provides a robust measure of how well our model is performing and produces a percentage figure

II. Analysis

Data Exploration

We will use the dataset provided by Udacity which is a large set of images, the dataset contains images of dogs and humans.

-Dog images dataset: The dog image dataset has 8351 total images which , The total number of images that were sorted on the train is 6,680 images, and test 836 images, and valid 835 images directories. Each of this directory(train, test, valid) have 133 folder corresponding to dog breeds.

-Human images dataset: The human dataset contains 13233 total human images , The images are different from each other, there may be more than one human face in one image, it may be 2 or 3 or more, and the backgrounds, shapes and colors also differ.
images from the dataset



Simple images from the dataset

Algorithms and Techniques

In the Algorithms and Technologies section, we will discuss the three main sections of our project:

To 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/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

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%.

```
Test Loss: 3.801174


Test Accuracy: 15% (132/836)
```

Screenshot of the results

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.

```
Test Loss: 1.835682


Test Accuracy: 64% (542/836)
```

Screenshot of the results

III. Methodology

Data Preprocessing

For processing we need input and in our model we need human images and dog images and each of them has classifiers as mentioned earlier to detect human images is done by pre-trained detectors to test the performance of the human face detector, for dog breed classifier we use images to train the data set so we need to do some Processing, including resizing the images to 224 * 224, and all images are straightened, and the image is enlarged to reduce over-processing.

Implementation

I have built a CNN network containing 5 convolutional layers, all convolutional layers have kernel size 3 and step 1, the first conversion layer (conv1) takes the image 224 * 224 and produces the final transform layer (conv5) with an output size of 128 using the RELU activation function, we have a connected layer Fully produce 133, pool layer (2,2) is used which will reduce the input volume by 2, add 0.20 leakage to avoid over processing.

Refinement

The CNN built from scratch has an accuracy of 15% although it meets the benchmarks, the model can be greatly improved by using transfer learning, I define the Resnet architecture as known for its excellent image classification performance, and the final fully connected layer is added to the fully connected layer With an output of 133.

IV. Results

Model Evaluation and Validation

Human Face detector: The human face detector function was created using OpenCV's implementation of Haar feature based cascade classifiers. 0.98 of human faces were detected in first 100 images of human face dataset and 0.17 of human faces detected in first 100 images of dog dataset.

Dog Face detector: The dog detector function was created using pre-trained VGG16 model. 1.0 of dog faces were detected in first 100 images of dog dataset and 0.0 of dog faces detected in first 100 images of human dataset.

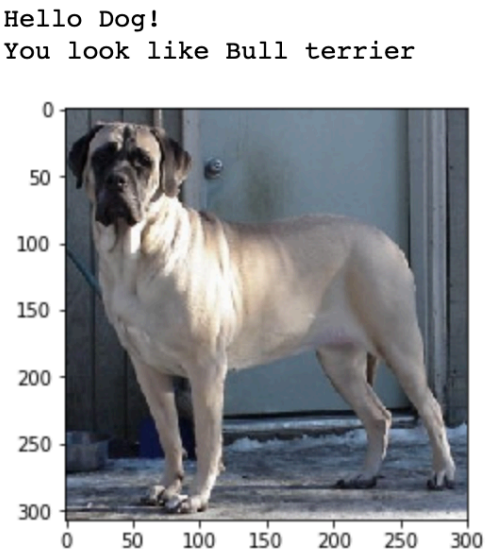
CNN using transfer learning: The CNN model created using transfer learning with ResNet50 architecture was trained for 15 epochs, and the final model produced an accuracy of 64% on test data. The model correctly predicted breeds for 542 images out of 836 total images.

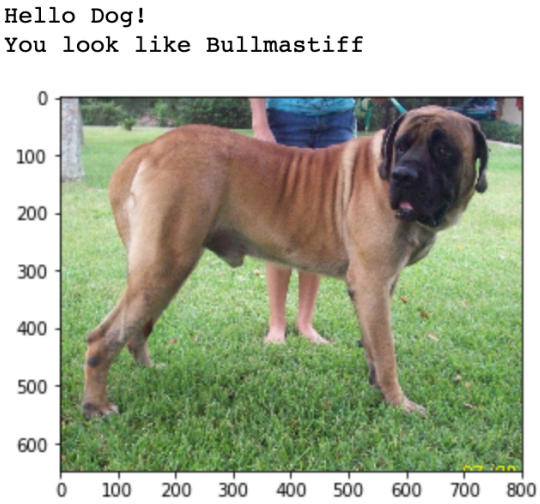
Justification

I think the model performance is very good, the model built using translational learning has an accuracy of 66% compared to the CNN model built from scratch with an accuracy of only 15%

V.Conclusion

Free-Form Visualization





Screenshots of the output

Reflection

Working on this project was very interesting. It required a lot of research and analysis for the best available solutions. It required several days and a lot of experiments, but the basic plan was implemented to achieve the desired goal.

Improvement

The model can be improved with other measures such as enlarging the image, improving accuracy, adding more training and testing data, and expanding the project to discover multiple dog breeds in the same image. In the picture, the number of breeds can be increased to more than 133.

References

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

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

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