Capstone Project Vinooj Kumar Naduparambil Viswanathan

Machine Learning Engineer Nanodegree June 5th, 2021

DOG BREED CLASSIFIER USING CNN

Domain Background

Dogs are one of the most common domestic animals and there are more than 200 breeds currently. It is essential to identify the dog breed to help differentiate their characteristics and select training methods based on the individual breed. Classifying the dog breeds from the image is tedious and a time-consuming process for large datasets, but with the help of Machine Learning and trained algorithms, it is much easier to predict the dog breed.

In this project, the most widely used Deep learning - Convolutional Neural Network is implemented using Transfer Learning to classify the dog breed.

Problem Statement

The end goal of the project is to build a ML model to process real-world, user-supplied images and predict the dog breed based on the input:

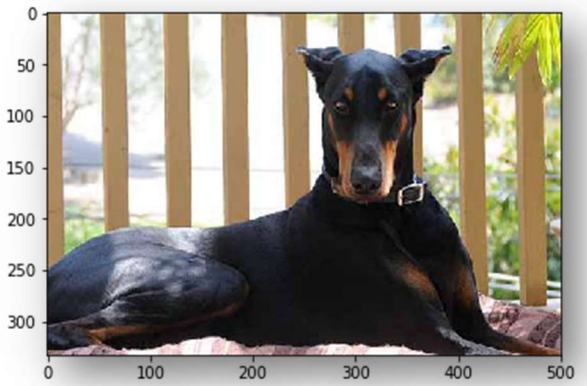
a. if a dog is detected in the image, return the predicted breed.
b. if a human is detected in the image, return the resembling dog breed.
c. if neither is detected in the image, provide output that indicates an error.

Datasets and Inputs

Human Dataset and Dog Dataset are downloaded and saved as numpy arrays. There are 13233 human images in Human dataset and 8351 dog images in Dog dataset.

Human Datasets – All the images are sorted in alphabetical order and of size 250x250. Data is imbalanced as some of the images contain more than 1 human face or half faces.

Dog dataset – Images are segregated into train (6,680 Images), test (836 Images) and valid (835 Images) folders each with 133 sub-folders corresponding to dog breeds. The dataset is imbalanced as the mean of images varies in each dog breed class.



Solution Statement

The standard procedure is to convert the images to grayscale before using any Face Detector. This image is then fed into the pre-trained face detector - OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces in images, which returns a Boolean value as output (TRUE if a human face is detected).

Load and transform the data into Train, Valid and Test Datasets. Selected pre-trained model VGG16, resized image size to 224x224 as VGG16 model requires input to be of 224x244 image size. Image augmentation is performed on all three datasets by random flipping, random rotation, and normalization of the given image.

Model architecture

1. Used 3 Convolutional layers in the CNN Architecture.
2. First 2 conv layers with kernel_size 3 and stride 2, which will downsize the input image by 2.
3. Next, the 3rd conv layer is used with kernel_size 3 and stride 1 and will not downsize the input image.
4. MaxPooling2D is used with kernel_size 2 and stride 2, which will further downsize the input image by 2.
5. Output from 3rd conv layer results in an image of size 128.
6. ReLU activation func is used and a Dropout layer of 0.25 is added to avoid overfitting.
7. Fully connected Linear Layer will produce an output image of size 133.

Selected **ResNet101** architecture as it has 101 deep layers which is best suited for Image Classification problem.

Steps:

1. Import ResNet101 pre-trained model.
2. To solve the classification problem, change the out_features of fully connected layer to 133.
3. Loss function selected- CrossEntropyLoss().

Benchmark Model : Evaluation and Validation

Evaluation Metrics

Classification Accuracy is the metric used to evaluate the performance of the algorithm. Equal number of images are provided to assess the accuracy of both Human Face detector and Dog detector algorithm. Test Loss and Accuracy is determined for Dog breed classifier.

i. CNN Model scratch: Benchmark Test Accuracy > 10%

Achieved validation accuracy as mentioned below by training the model with 15 epochs on the test dataset of dog images.

- Test Loss: 3.714258
- Test Accuracy: 13% (116/836)

ii. CNN Model using Transfer Learning: Benchmark Test Accuracy > 60%

Achieved validation accuracy as mentioned below by training the model with 10 epochs on the test dataset of dog images.

- Test Loss: 0.546267
- Test Accuracy: 83% (701/836)

Project Design

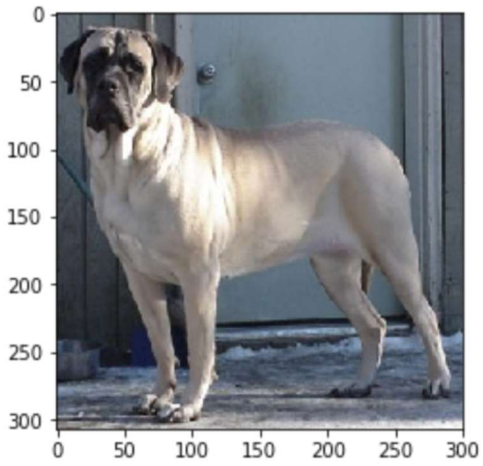
Following tasks are involved:

- Download and import Dog and Human Datasets.
- Detect Human faces in images using Open CV's implementation of Haar feature-based cascade classifiers.
- Import pre-trained TorchVision and VGG-16 models and implement a Detect Detector.
- Create a CNN to Classify Dog Breeds (from Scratch) with at least 10% test accuracy.
- Create a CNN to Classify Dog Breeds (using Transfer Learning) with at least 60% test accuracy.
- Implement the Algorithm to accept a file path to an image and determine whether image contains a dog, human or neither. Then,
    d. if a dog is detected in the image, return the predicted breed.
    e. if a human is detected in the image, return the resembling dog breed.
    f. if neither is detected in the image, provide output that indicates an error.

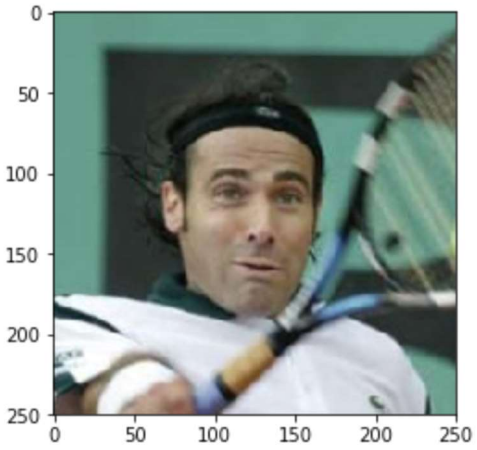
Reflection

The algorithm performs as expected. Given any image, the algorithm predicts an estimate of the canine's breed, if a dog image is detected. If supplied an image of a human, the code identifies the resembling dog breed.

```
Hello, Doggy !!
The Dog in the image belongs to breed : Bullmastiff
```




```
Hello, Human !!
The Human in the image resembles : Irish red and white setter
```



Improvement

Few areas to improvise the algorithm –

- Utilize accurate model for human face detection.
- Improve Training models with huge and complex data (for ex. Human and Dog in an image).
- Hyper parameter tuning to enhance performance.
- Augmentation trials to increase accuracy.

References

- GitHub Repository: https://github.com/udacity/deep-learning-v2- pytorch/blob/master/project-dog-classification/

- Resnet101: https://pytorch.org/vision/stable/_modules/torchvision/models/resnet.html#resnet101

- Transfer Learning: https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html

- GitHub Imagenet: https://github.com/pytorch/examples/tree/master/imagenet

- PyTorch Documentation: https://pytorch.org/docs/master/

- CNN guide: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neuralnetworks-the-eli5-way-3bd2b1164a53