Machine Learning Engineer Nanodegree Udacity

Capstone Proposal

Vishal Sharma 21st, 2021

Dog Breed Classifier

Domain Background

The requirement of this project is the classification of the different dog breeds. In our world, from east to west and north to south exists a lot of breeds of dogs that differ one to another by a lot of parameters such as size, height, weight, type of employment, type of intelligence, etc.

For our project, the classification is absolutely based on images of the dogs and the goal defined for this is to classify 133 dog breeds using state of art deep learning algorithms to teach and train the computer how to give an estimation of a particular dog breed from a dog image provided.

Image classification is one of the most vibrant and dynamic field in Machine Learning, computer vision and one other very important challenge is the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) where each year review of the new classification algorithms[1].

Problem Statement

The ultimate purpose of this project is to write an algorithm that accepts a file path to an image and first determines and detects whether the image contains a human, dog, or neither.

After a detection process, we could have the following:

1. Dog detection in the provided image, return the predicted breed.
2. Human detection in the provided image, return the resembling dog breed.
3. Neither detection in the provided image, provide an output that indicates an error.

Datasets and Inputs

The datasets required for the project are two: dogs dataset and human dataset.

The dogs dataset is composed by following :

- Training: 6680 images
- Validation: 835
- Test: 836
- Total images: 8351
- Classes: 133

The human dataset is composed of 13233 human images. First of all, images are resized to 244x244 and then normalized before being used by the model..

Solution Statement

There are 7 main steps for the solution of the project as follows.

Step 1: We need to explore and check the datasets in order to understand how to use them and choose the proper algorithms for this.

Step 2: Then Implementing a Haar feature-based cascade classifier by using OpenCV in order to detect faces in the human dataset provided.

Step 3: Then I will use a pre-trained VGG16 model so that it can detect dogs in the dogs dataset provided.

Step 4: Then I will try to create a LeNet[2] like architecture that uses CNN in order to classify the 133 dogs' breeds and have an accuracy of more than 10% as required.

Step 5: I will use the transfer learning technique in order to get a pre-trained ResNet50 architecture and then continue the training with the dog's dataset provided. The minimum required accuracy is 60% in the test set given.

Step 6: I will write a custom algorithm that accepts a file path for the image and first of all it will detect whether the image provided contains a human, dog, or neither.

Step 7: And also I will Test the Algorithm with some random images found online.

Benchmark Model

This model will be compared with various benchmark models in a Kaggle competition (https://www.kaggle.com/c/dog-breed-identification/discussion).

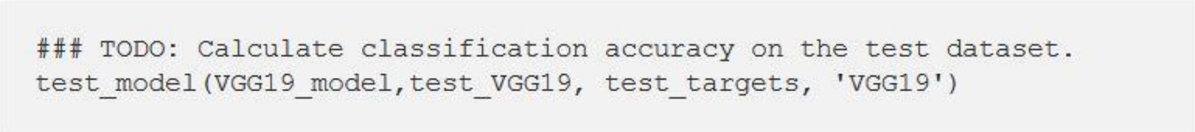
To tackle this complex data, we use a benchmark model to build a quite basic pipeline and a well-versioned model to improvise our classification accuracy rate. Such processes are carried to tune our model for better and accurate prediction of results. The benchmark model gives us a view to make a comparison and reduce the overfitting or underfitting condition and tune up the model for an optimized outcome. Logistic Regression, KNN are such good examples of the better benchmark. We will also use the predefined image classifiers for example ImageNet, ResNet, VGG16, etc. to classify our images and later optimize our pipeline for the accurate evaluation of metrics.

In respect of Dog Identification work in Kaggle, we can see that "Mohamed Sheik Ibrahim" used theVGG19, a predefined base model, and carried different processing techniques such as data augmentation to increase and improve the results obtained from the predefined model. He also used the logistic regression technique for the classification of the images of dogs and achieved an accuracy of 68%.

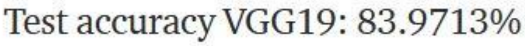
Let see one more scenario by considering work performed on the same data, using the inception v3, the pre-trained model for image classification, Carey B achieved an accuracy of 83%, which is considered to be a quite good classification rate based on the performance of the model.

Using the all above understandings considering the result, I will be using VGG16 for our data for classifying the breeds of the Dogs, Later I will build a CNN (Convolutional Neural Network) and tune the parameters as required, And then by using transfer learning through these models, I will make a comparative study and analyze the performance and accuracy of the model.

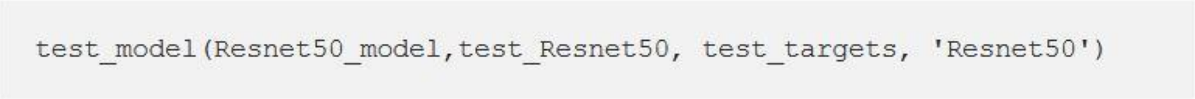
The figure below is the accuracy results of some pre-trained models created earlier. I will consider these values to benchmark the performance of our model.



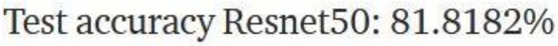
```
### TODO: Calculate classification accuracy on the test dataset.
test_model(VGG19_model,test_VGG19, test_targets, 'VGG19')
```



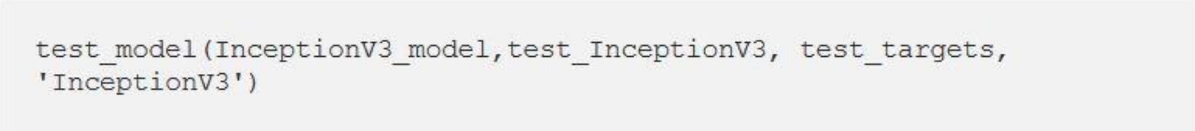
Test accuracy VGG19: 83.9713%



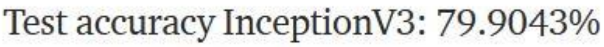
```
test_model(Resnet50_model,test_Resnet50, test_targets, 'Resnet50')
```



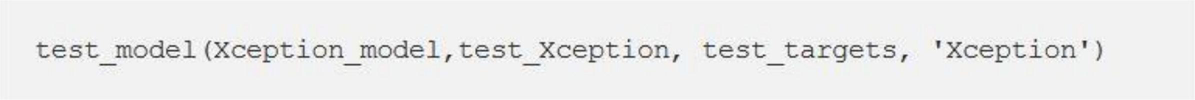
Test accuracy Resnet50: 81.8182%



```
test_model(InceptionV3_model,test_InceptionV3, test_targets,
'InceptionV3')
```



Test accuracy InceptionV3: 79.9043%



```
test_model(Xception_model,test_Xception, test_targets, 'Xception')
```

Evaluation Metrics

In order to work and deal with a multi-class classification problem, the negative log-likelihood loss function will be used as the evaluation metrics.

Using the negative log-likelihood loss function as evaluation metrics, the algorithm will calculate each iteration and the distance of a predicted output to the corresponding label.

In this manner, the algorithm will learn from it and it will adjust the predictions in order to minimize this distance (that is loss).

Project Design

The solution designed for this project will follow these steps:

Step 1: Datasets exploration and check.

Step 2: Determines and detect Humans using a Haar feature-based cascade classifiers

Step 3: Determines and detect Dogs using a pre-trained network

Step 4: Create a CNN for Classification of Dog Breeds (from Scratch) using a LeNet like architecture.

Step 5: Create a CNN for Classification of Dog Breeds using Transfer Learning and using a ResNet50 architecture.

Step 6: Write a custom Algorithm that accepts a file path to an image and first of all it will determine whether the image contains a human, dog, or neither.

Step 7: Test the Algorithm with some random sample images found online.

Reference

1. https://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html

2. Olga Russakovsky*, Jia Deng*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. (* = equal contribution) ImageNet Large Scale Visual Recognition Challenge. IJCV, 2015

3. https://en.wikipedia.org/wiki/LeNet

4. https://hackernoon.com/a-brief-history-of-computer-vision-and-convolutional-neural-networks-8fe8aacc79f3

5. https://medium.com/nanonets/how-to-easily-build-a-dog-breed-image-classification-model-2fd214419cde