Capstone Project Report

Sakthivel Rathinasamy

Machine Learning Engineer Nanodegree **01-May-2020**

Definition

Project Overview:

- The Dog identification app is one of the default project suggested by the Udacity, which helps in learning and building skills in machine learning.
- The problem is to identify a dog if dog image is given as input, if supplied an image of a human, we have to identify the resembling dog image.
- The object is to build the model and the app that can process real world user input images and identify an estimate of the dogs breed.
- This is a multi-class classification problem where we can use supervised machine learning to solve this problem.

Problem Statement :

Objective is to build a machine learning model that can be used within web app to process real-world, user-supplied images. The algorithm has to perform two tasks:

- ***Dog face detector:*** Given an image of a dog, the algorithm will identify an estimate of the canine's breed.

- ***Human face detector:*** If supplied an image of a human, the code will identify the resembling dog breed.

- **Datasets and Inputs** For this project, the input format must be of image type, because we want to input an image and identify the breed of the dog. The dataset for this project is provided by Udacity. The dataset has pictures of dogs and humans.

Metrics:

Metrics of this realtime scenario is usually derived using the accuracy metrics, because of the imbalance in the dataset, accuracy is a not a good indicator here to measure the performance.

We have train, test and valid datasets. Once the model is trained using train dataset, it will be tested against the test datasets for the performance.

Log loss takes into the account of uncertainty of prediction based on how much it varies from actual label and this will help in evaluating the model.

Analysis

Data Analysis:

Dog images dataset:

https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/dogImages.zip

Below are the dog images split up
test Images : 836
valid Images :835
train Images : 6680
Total Images : 8531

Dog images varies with size and background.

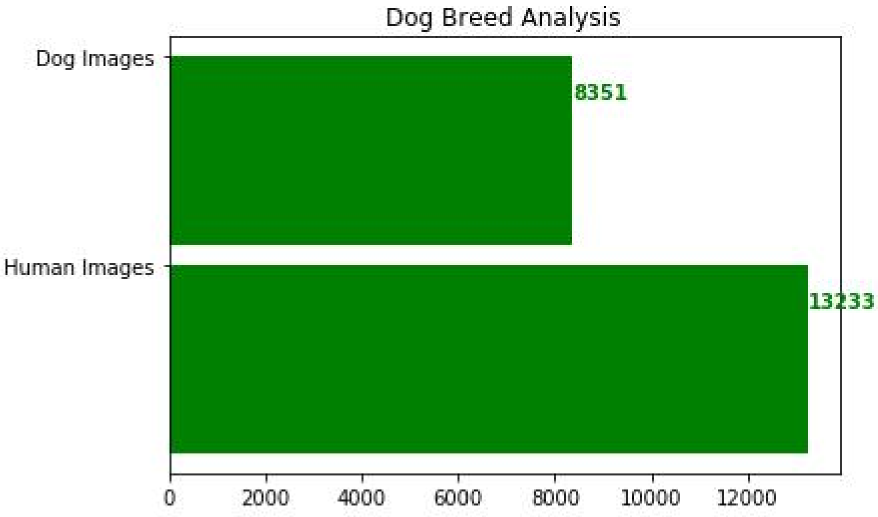
Human images dataset:

https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/lfw.zip

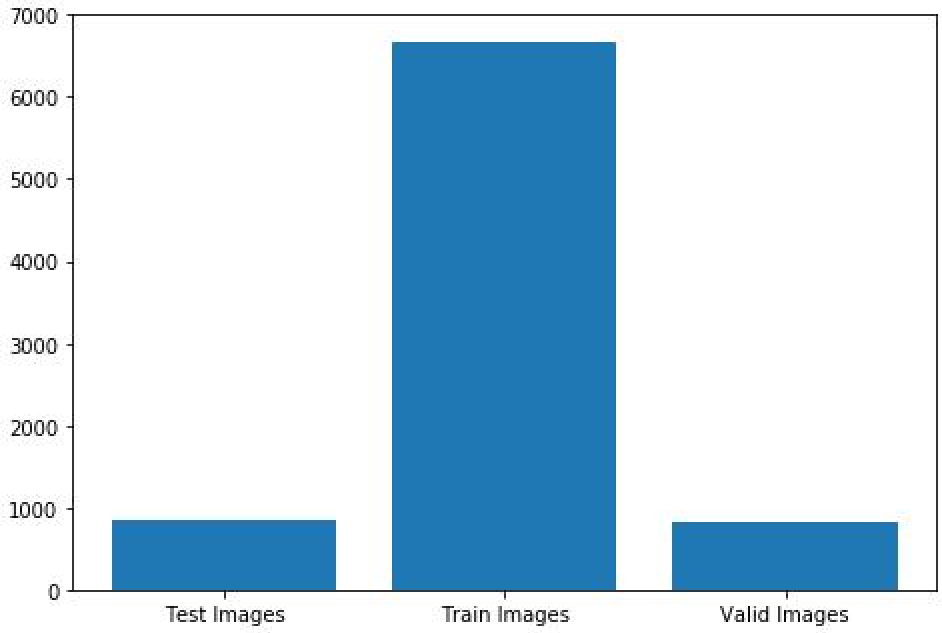
Total Images : 13233
All the images are 250X250 pixel size and 96 dpi resolutions.
Total Folders : 5750
Single image can have one human face or multiple human face as well.

Exploratory Visualization:

Overall Dataset splitup:



Total Dog dataset splitup:



Algorithims and Techniques:

Convolutional Neural Network is a type of artificial neural network used in image recognition and processing that is specifically designed to process large pixel data. Neural Networks mimic the way our nerve cells communicate with interconnected neurons and CNNs have a similar architecture. In our project for solving the problem we will be using CNN Object Detection using ***Haar feature-based cascade classifiers*** is an effective object detection method.It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. In our case have both dogs and human images to test.

- Step 1: Human Face detector using the OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces in images.
- Step 2: Dog Face dectector using the VGG-16 model, along with weights that have been trained on ImageNet.
- Step 3: Create a CNN to Classify Dog Breeds from scratch
- Step 4: Create a CNN to Classify Dog Breeds using transfer learning

Benchmark:

Model created from scratch and Model created from the transfer learning was used. The test accuracy of this two models output have provided the best performing model or approach. This way comparision between two models have been done. The results of the models are available in the Result section of this document. As part of our project requirement the test accuracy should be above 60%. I was able to get the better performance from one of the model. This shows my model is performing better than the project benchmark.

Methodology

Data/Image Preprocessing :

For Uniformity all the data/images are resized 224 X 224, this would help in data processing. Also normalization technique is applied to all the data sets train, test and valid.

To reduce the overfitting, image augmentation is done. Images are converted to tensor finally.

Below are the code snippet:

```
transformations = transforms.Compose(
[transforms.Resize([224, 224]),
transforms.ToTensor(),
transforms.Normalize(mean=[0.485, 0.456, 0.406],
std=[0.229, 0.224, 0.225])])
```

Implementation:

Step 1: Import the necessary dataset and libraries, Pre-process the data and create train, test and validation dataset.
Step 2: Human Face detector using the OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces in images
Step 3: Dog Face detector using the VGG-16 model, along with weights that have been trained on ImageNet.
Step 4: Create a CNN to classify dog breeds from scratch, train, validate

and test the model.
Step 5: Create a CNN to Classify Dog Breeds using Transfer Learning ,train and test the model.
Step 6: Dog breed classification algorithm :

i. If a dog is detected , return the predicted breed.
ii. If a human is detected , return the resembling dog breed.
iii. If neither is detected , provide output as other thing.

Step 7: Validate the algorithm using sample images and test.

Refinement:

Object Detection using Haar feature-based cascade classifiers is an effective object detection method.It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. In our case have both dogs and human images to test.

Human Face detector using the OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces in images.

Human face detected in 100 images of Human dataset : 98%
Human face detected in 100 images of Dog dataset : 17%

Dog Face detector using the VGG-16 model, along with weights that have been trained on ImageNet.

Dog face detected in 100 images of Dog dataset : 100%
Dog facce detected in 100 images of Human dataset : 2%

Create a CNN to Classify Dog Breeds from scratch

Model Scratch Test Loss : 4.7%
Model Scratch Test Accuracy : 5%.

Create a CNN to Classify Dog Breeds using transfer learning

Model Transfer Learning Test Loss : 0.86%.

Model Transfer Learning Test Accuracy : 76%.

Result

Model Evaluation and Validation:

Model Transfer Learning works better compare to the Model Scratch, as the transfer learning is trained with large base of images. This makes the classification of the breed easier and faster. Below are the metrics for your review

Model Scratch Test Loss is 4.7%
Model Scratch Test Accuracy from the scratch is 5%.

Model Transfer Learning Test Loss is 0.86%.
Model Transfer Learning Test Accuracy is 76%.

Justification:

Based on the observation the model transfer learning is performing better compared the model scratch. Model transfer learning has the accuracy of 76% in 5 Epochs.

Conclusion

Output Results from the validation app:

Below are the output results of the running the app to test the model transfer learning:

Predicted breed: Boykin spaniel

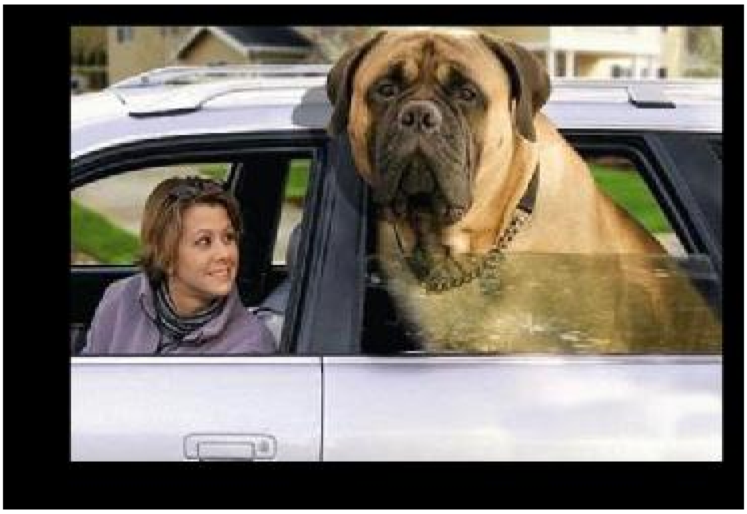
Predicted breed: Glen of imaal terrier

Predicted breed: Akita

Predicted breed: Mastiff

Predicted breed: Mastiff

Predicted breed: Mastiff



Reflection:

The process used for this project can be summarized using the following steps:

Step 1: Import the necessary dataset and libraries, Pre-process the data and create train, test and validation dataset.
Step 2: Human Face detector using the OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces in images
Step 3: Dog Face detector using the VGG-16 model, along with weights that have been trained on ImageNet.
Step 4: Create a CNN to classify dog breeds from scratch, train, validate and test the model.
Step 5: Create a CNN to Classify Dog Breeds using Transfer Learning ,train and test the model.
Step 6: Dog breed classification algorithm :
Step 7: Validate the algorithm using sample images and test.

Challenging part was on the Step 3, 4, 5.

Was changing the batchsize parameters of the dataloader and validating whether the accuracy is increasing or not, i have to do it for multiple times to get the optimal batchsize.

Improvements:

- Overall in the world we have 340 dog breeds. Currently, we use only 133 breeds. If we have more breeds and images, would help to train the model with more training datasets.

- I have changed the batchsize to 22. Need to change the batchsize to 50 or more to see the performance and accuracy.
- We should try the training using different algorithm.

References :

https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html

https://www.kaggle.com/c/dog-breed-identification

https://www.mathworks.com/help/deeplearning/ref/resnet101.html

https://www.programcreek.com/python/example/104830/torchvision.transforms.Normalize

https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035