Machine learning Engineer nanodegree

Capstone Project report

Dog Breed Classification with CNN

1.Definition

Project Overview:

As many people now days are seeking for buying a dog and are new to this business the first thing to know is the breed of the dogs in order to know which one will suit them the most and their environment and know more about their behavior therefor the overview of this project is to build a model which is used to identify the breed of a dog by just giving an image to the model as an input and it will classify it by extracting the features of the dog then detecting the breed of the dog . If the model is supplied with a human image it has to identify the resembling dog breed.

This model will be so useful for dog owners as it will prevent them from getting scammed and it will also save them lots of time and money for identifying the breed of their dog .

Problem Statement

Building a pipeline to process real world, user supplied images that can be used in an application .The model has to perform two tasks.

Human face detection : If the model is given a human image it has to identify the resembling dog breed.

Dog detection : when the model is given an image of a dog it has to detect it’s breed

Metrics:

I am going to take in consideration the accuracy as an evaluation metrics and as the benchmark model specifies the accuracy

Accuracy = correct specification /all specifications

2.Analysis

Data Exploration

The dataset provided by Udacity contains images of dogs and humans

The dog images datasets : (8151 image) of dogs with different sizes and backgrounds which are divided into training , testing and validation.

Training (6,680) image , testing(836) image and validation (835) image . The data is also divided to directories with 133 folder each . The number of images of each breed varies widely which makes the data unbalanced .

The humans images datasets : (13233) images of humans all with same size of 250X250 but the pictures are taken of different humans and different angles which also makes the data unbalanced.

Algorithms and Techniques

The algorithm used for solving this problem is Convolutional Neural Network (CNN) and we used various techniques each for specific purposes the first is the pretrained VGG-16 model which I used to recognize the dogs in the images and it is only used for prediction and will not be trained then I used the

Cascade Classifier from OpenCV to detect the faces of humans then I have used PyTorch to implement ,train and test the model to detect the breed of the dog and lastly I have used transfer learning as I have used Relu .

Benchmark

I have taken the VGG-16 model as a benchmark model which is Very Deep Convolutional Networks for large Image Recognition. The model achieves 92.7% top 5 test accuracy in ImageNet , which is a dataset of over 14 million images belonging to 1000 classes.

The model created from must have an accuracy of 60% or more.

3.Methodology

Data Preprocessing

As the images were having different resolutions which made the data a little bit imbalance I used PyTorch module torchvision.transforms and reduced the resolution to 224X224 then I applied normalization to the train , valid and test datasets and augmented the training data to reduce overfitting and underfitting .The training data was randomly rotated and randomly horizontally flipped then converted the images to tensor.

Implementation

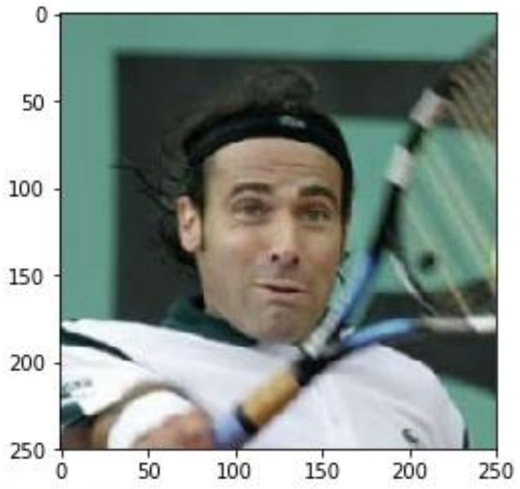
I have build a Convolutional Neural Network from scratch following the VGG-16 architecture and I made it with only 3 layers to avoid complexity each with kernel size 3 and stride 1 with the first taking an input of 224X224 images and the third producing an output size of 228 and a pooling layer of kernel size 2 then I activated the Relu function.

Regarding the human face detection I have used the Cascade Classifiers to check whether a person's face is included in the image or not ,I have used the cascade classifier of haarcascade_fronalface_alt.xml file and implemented the face_detector function as it is able to recognize 100% of humans as humans**.**

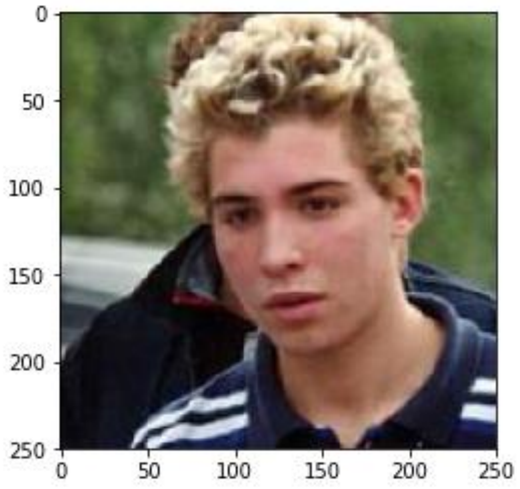
Refinement

In the dog breed classifier with transfer learning I have used REsNExt-50 with accuracy of 71% .The CNN model created form scratch accuracy was 13%

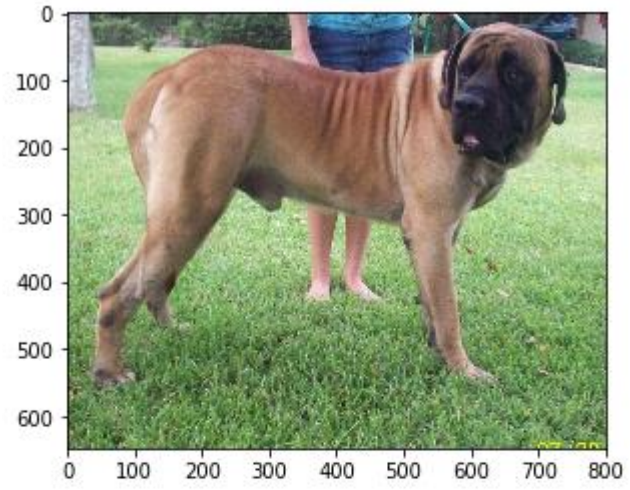
```
Detected Human!
Predicted breed:  English toy spaniel
```



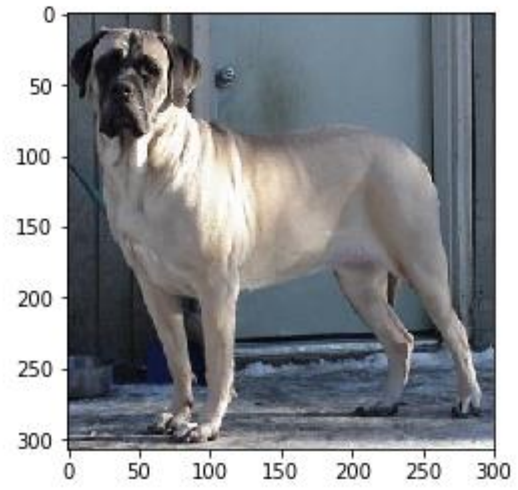
```
Detected Human!
Predicted breed:  Petit basset griffon vendeen
```



```
Detected Dog!
Predicted breed:  Cane corso
```



```
Detected Dog!
Predicted breed:  Bullmastiff
```



```
Detected Dog!
Predicted breed:  Bullmastiff
```

4.Results

Model Evaluation and Validation

Dog face detector : The dog face detector function was created using VGG-16 model 100% of dog faces were detected in the first 100 images of dog dataset and 1% of dog faces detected in first 100 image of human dataset.

Human face detector :For implementing the human face detector function I used the Haar feature based cascade classifier form python OpenCv 98% of human faces were detected in the first 100 images of human face dataset and 17% of human faces detected in first 100 images of dog dataset.

CNN using transfer learning : the model is created using transfer learning with ResNet-50 architecture and the fi nal model accuracy was 71% on test data 597/836 and t he Test Loss: 0.958761.

Justification

For me the performance of my model was great as the accuracy of the transfer learning model is 71% and the CNN model was 13%

5.Conclusion

Reflection

1.Identifying the problem and looking for releavant datasets

2.Downloading the dataset

3.Implementing the human face detector

4.Implementing the dog face detector

5.Implementing the CNN model from scratch

6.Implementing the transfer learning model and training and testing it

7.Documenting the project

Improvement

The model could be improved by increasing the convolutional layers but I have avoided complexity and the algorithm could also be improved by increasing the dataset and hyperparameter tunning and one of the methods also that could have improved the algorithm was implementing ResNet-101 which is very deep neural network.

References

https://pytorch.org/docs/stable/index.html

https://github.com/Abdelrahman13-coder/Capstone-project-Dog-Breed/blob/master/dog_app.ipynb

https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html

https://www.youtube.com/channel/UCxxljM6JkSvJVSD_T90ZnMw