Machine Learning Engineer Nanodegree

Capstone Project

Youssef Lasheen | 4/2020

Definition

Project Overview

In this project, Given an image of a dog, the algorithm will identify an estimate of the canine's breed. If supplied an image of a human, the code will identify the resembling dog breed.

Problem Statement

The aim of the project is to build a pipeline to process real-world, user-supplied images. The algorithm will identify an estimate of the dog's breed given an image. When the image is of a human, the algorithm will choose an estimate of a dog breed that resembles the human. If neither a dog or a human is detected, then an error message is output. Therefore, the models in place should be capable of detecting a dog or human in an image, classify the dog to its breed and classify a dog breed that the human resembles.

Metrics

Accuracy will be the main metric used to test both the benchmark model and the solution model. **Accuracy = (T P +T N) (T P +T N+F P +F N)**

True Positive(TP), True Negative(TN), False Positive(FP) & False Negative(FN).

Analysis

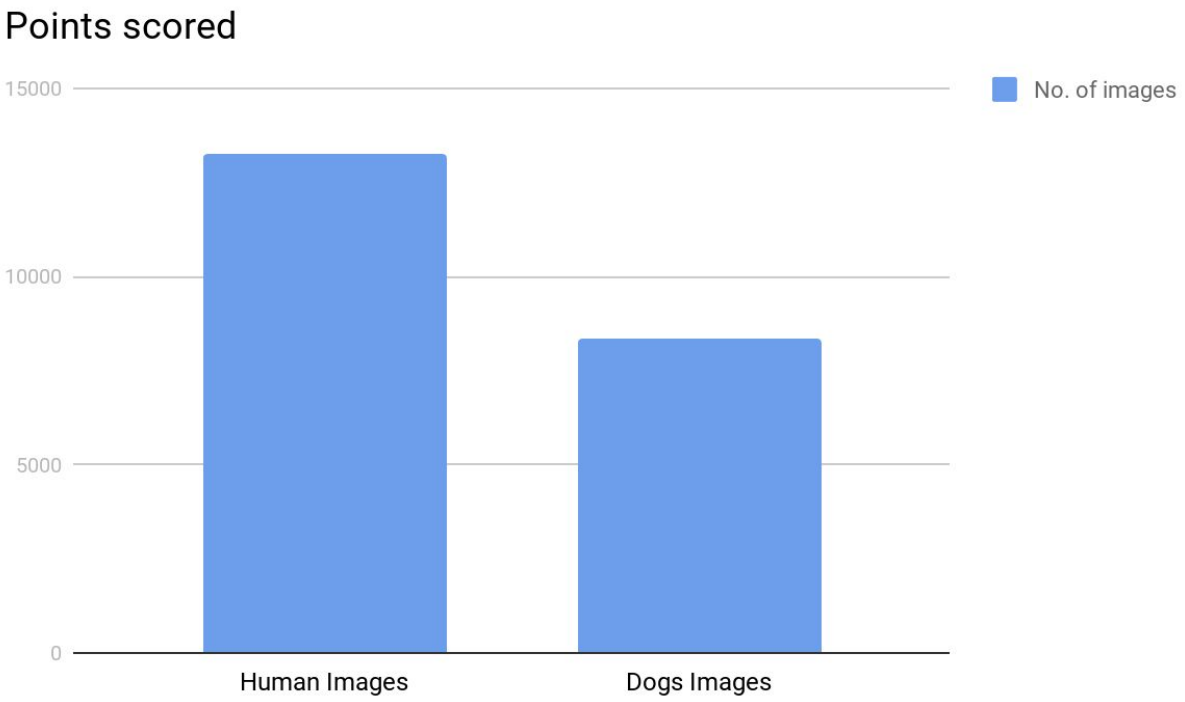
Data Exploration

Datasets used to train, test and validate the model are provided by Udacity. There are a total of 13233 human images and 8351 dog images.

- **Dogs Dataset:** There is the dog dataset containing 133 breeds and each breed contains 8 images. Which is used to detect dogs in images using a pre-trained VGG-16 model.
- **Humans Dataset:** There is also the human dataset which contains images, first names and last names. That will be used to detect human faces in images using OpenCV's implementation of Haar feature-based cascade classifiers.

Exploratory Visualization

The plot below shows the distribution of provided datasets:



Algorithms and Techniques

To perform this classification, we can use CNN to solve the problem. A CNN is a Learning algorithm which can take in an input image, assign learnable weights and biases to the image.

The solution involves three steps.

1- Detect human images, we can use existing algorithms like OpenCV's implementation of Haar feature based cascade classifiers.

2- Detect dogs images and we will use a pretrained VGG16 model.

3- pass this image to an CNN model which will process the image and predict the breed that matches the best out of 133 breeds.

Benchmark

Create a CNN to Classify Dog Breeds Section (using transfer learning):

I tried different architecture such as alexnet, resnet18 and vgg-19. Neither of them gave me better accuracy than resnet50. I do not want to use inception or resnet higher versions as I felt these pretrained layers were already trained on dog breeds.

Methodology

Data Preprocessing

Create a CNN to Classify Dog Breeds Section (from scratch and using transfer learning):

-A RandomResizedCrop is applied to stretch the image to 224:244 px. I used this because this is what is used for the VGG16 architecture, that performs at an accuracy higher than 10%.

-Since there are different kinds of dogs with similar looks, we have to choose the correct augmentation. I decided to augment the dataset by using RandomHorizontalFlip and RandomRotation by 10 degrees in an attempt to increase the models ability to generalize.

Implementation

The model has 3 convolutional layers. All convolutional layers have kernel size of 3 and stride 1.

The final layer produces an output size of 128.

A dropout of 0.30 is added to avoid overfitting.

Refinement

The model created from scratch has accuracy of 14%, The model can be significantly improved by using transfer learning.

To create CNN with transfer learning, I have selected the Resnet50 architecture which is pre-trained on ImageNet dataset.. The model performed extremely well when compared to CNN from scratch., the model got 86% accuracy.

Results

Model Evaluation and Validation

Create a CNN to Classify Dog Breeds Section (from scratch):

As I am only required to achieve a 10% accuracy I made a simple model. I chose to have 3 conv layers with a kernel size of 3x3 and a padding of 1 because the image size is 224 which is not divisible by kernel size of 3 so padding by 1 will make the image to 225x225. I used dropout to avoid overfitting as going deeper we might overfit.

Create a CNN to Classify Dog Breeds Section (using transfer learning):

I chose resnet50 and I updated the final layer to take inputs and produce 133 breeds as an output.

Resnet has trained layers without feature extraction layers or convolutional layers. This model is suitable because the dog breed problem deals with RGB images and ResNet has 3 channels. So it makes sense. Also resnet have skip connections in them which helps to prevent overfitting when training.

Justification

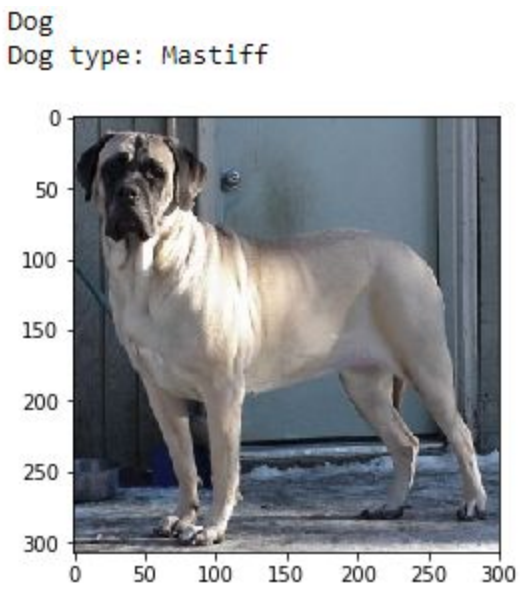
Create a CNN to Classify Dog Breeds Section (using transfer learning):

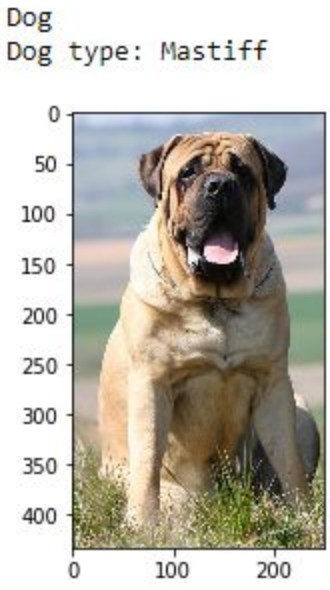
I tried different architecture such as alexnet, resnet18 and vgg-19. Neither of them gave me better accuracy than resnet50. I do not want to use inception or resnet higher versions as I felt these pretrained layers were already trained on dog breeds.

Conclusion

Free-Form Visualization

Example of the output of the algorithm.





Reflection

Detect humans section:

It used OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces from the images supplied.

Detect dogs section:

A pre-trained VGG-16 model with trained weights on ImageNet, a large popular dataset for image classification, will be used to detect dogs in the user-supplied images. The model should be checked that it returns indexes from 151 to 268 inclusive, that is, it must include categories from 'Chihuahua' to 'Mexican Hairless'.

Create a CNN to Classify Dog Breeds Section (from scratch):

A CNN model is then built from scratch to classify dog breeds, that is, transfer learning cannot be used just yet. This model should surpass a test accuracy of 10% set by Udacity because the model is being built from scratch so classifying similar breeds can be a challenge however transfer learning will greatly improve this.

Create a CNN to Classify Dog Breeds Section (using transfer learning):

A transfer learning will be used with a ResNet50 model to significantly boost the accuracy of the CNN model. It should surpass the 60% test accuracy set by Udacity.

Improvement

- Identify multiple dogs with different breeds

- Print confidence level to the user

- Hyper-parameter tuning and training