MACHINE LEARNING ENGINEER NANODEGREE

DOG BREED CLASSIFIER WITH CONVOLUTIONAL NEURAL NETWORK

John Nikhil Kollanoor

20th SEPTEMBER 2020

Project Overview

The problem is to identify the breed of the dog, given image of a dog as input or identify the closest resembling dog breed in case of image of a human. A pipeline is to be built that processes real world images as input and returns the name of the breed. The problem is multi class classification as multiple breeds need to be successfully identified. Supervised machine learning model is used to solve this problem.

Problem Statement

The goal of the project 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.

The goal is to build a ML model that can be implemented as an application to process a real-world image given as input and do the following

- Detect if a dog is in the image, if yes predict the breed
- Detect if a human face is in the image, if yes predict the closest resembling breed

The above goal is achieved in two ways

- Building a CNN model from scratch using the images provided for training.
- Using transfer learning, where a pretrained model is downloaded and used.

Metrics

A good classification model should successfully identify the class of the input. There are numerous metrics to evaluate how good a model is. Some of them are Precision, Recall, F1 score etc. I chose accuracy as my metric. This would give the ratio of successfully identified classes (in this case breed) to total predictions.

During the training of model, test data prediction is compared with validation dataset and multiclass log loss is calculated to find the best performing model. Log loss accounts for uncertainty in prediction based on differences in prediction and label.

Data Exploration

The input is given as an image to simulate real world image processing application. The dataset consists of images of dogs and humans.

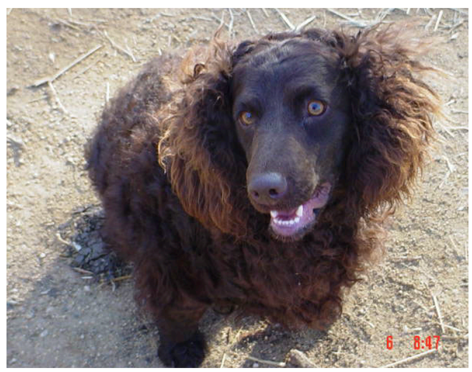
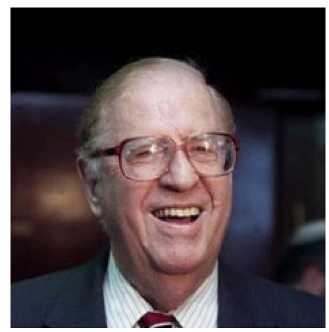
Dog images dataset:

The dog image dataset has 8351 total images which are split into train (6,680 Images), test (836 Images) and valid (835 Images) directories. Each of these directories (train, test, valid) have 133 folders corresponding to the different dog breeds. The images are of different sizes and different backgrounds and are not full sized (some of the subjects are cropped). The data is also not balanced because the number of images provided for each breed isn't the same.

Human images dataset:

The human dataset contains 13,233 total human images which are split into different folders based on the names of human (5750 folders). All images are of size 250x250. These images also have different backgrounds and subjects captured in various angles

Sample Images:

Algorithms and techniques

Convolutional neural network is a c lass of deep neural networks, most commonly applied to analyzing visual imagery. A CNN model is built from scratch and also used through transfer learning to solve the problem in two different ways. To detect human images, existing algorithm like OpenCV's implementation of Haar feature based cascade classifiers is used. To detect dog-images a pretrained VGG16 model is used. Finally, after the image is identified as dog/human, the image is passed to an CNN model which will process the image and predict the breed that matches the best out of 133 breeds.

Benchmark

The benchmark accuracy for the CNN model should be at least 10%. This can confirm that the model is effective with significance. This is because a random guess corresponds to almost 1% accuracy (1 in 133 chance of guessing the correct breed).

Data Preprocessing

Images are resized to 224 x 224 size as that is the standard input of image size VGG model takes. Image augmentation is also implemented to reduce overfitting. Other transformations are applied such as random rotations and horizontal flips. The images are converted into tensors using pyTorch to pass to the model.

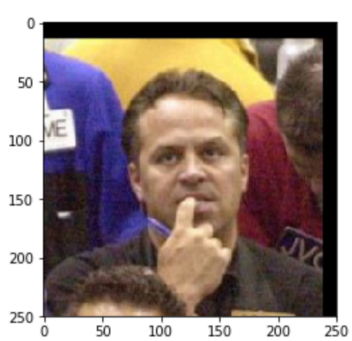
Implementation

The CNN model has 3 convolutional layers. All convolutional layers have kernel size of 3 and stride 1. The first conv layer takes the input image with size 224*224 and the final conv layer produces an output size of 256. ReLU activation function is used. The pooling layer of (2,2) is used which will reduce the input size by 2. We have two fully connected layers that finally produces 133-dimensional output. A dropout of 0.3 is added to avoid over overfitting.

Refinement

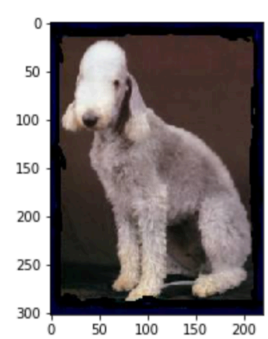
The CNN built from scratch (trained with the images that were provided) has an accuracy of 14%. The model can be improved even though it meets the benchmark using transfer learning. Resnet 101 architecture is pretrained on ImageNet dataset with 101 layers. The output of Resnet 101 is fed as input to the model that was built from scratch. This is achieved by adding a fully connected layer to produce 133-dimensional output (one for each dog breed). The model has performed well compared to the model that was built and trained from scratch achieving accuracy of 82%

```
Human detected
```



```
Predicted closest resembling dog breed: Australian shepherd
```

```
Dog detected
```



```
Predicted breed: Bedlington terrier
```

Model Evaluation and Validation

Human Face detector:

The human face detector function was created using OpenCV's implementation of Haar feature based cascade classifiers. 98% of human faces were detected in first 100 images of human face dataset and 17% of human faces detected in first 100 images of dog dataset.

Dog Face detector:

The dog detector function was created using pre-trained VGG16 model. 94% of dog faces were detected in first 100 images of dog dataset and no dog faces were detected in first 100 images of human dataset.

CNN using transfer learning:

The CNN model created using transfer learning with ResNet101 architecture was trained for 5 epochs, and the final model produced an accuracy of 82% on test data. The model correctly predicted breeds for 686 images out of 836 total images. Accuracy on test data: 82% (686/836)

Justification

The model which was created using transfer learning has achieved 82% accuracy which is considered to be better than the CNN model that was built from scratch which yielded 14% accuracy.

Improvement

The transfer learning model outperforms the model built from scratch.

The model built from scratch is not complex enough to capture variances to detect dog breed and hence it requires the following to improve

- More volume of training data and training with a deeper network.
- Hyper parameter tuning to optimize and find out the best possible solution.
- More image manipulation techniques such as augmentation, filters can be tried to improve accuracy.