Capstone Project

Mohammed Azeem
26th June, 2020

Machine Learning Engineer Nanodegree

Dog Breed Classifier using Convolutional Neural Network

Domain Background

Dog breeding is a known problem in ML. The problem is identifying a dog breed if a dog's image has been assigned as an input, and when providing a human image, it should identify the same type as the one made for the dog. The idea is to create a

pipeline that can process real-world images provided by the user and identify caninetype equations. This is a multi-level problem where we can use machine learning to be monitored to solve this problem. After completing this model, I plan to build a web app where the user can upload an image and receive predictions from the model. This project gives me the opportunity to create and export ML models, so I chose this as my capstone project

Problem Statement

The goal of this project is to build a machine learning model that can be used within web app to process real-world, user-supplied images. The algorithm performs these tasks:

Dog face detector: Given an image of a dog, the algorithm tries to estimate the breed of the dog.

Human face detector: When an image of a human is supplied , the code will identify the closest resembling dog breed.

Metrics

Data is segmented by train, test and valid data. The model is trained using train data. We use experimental data to predict model performance on abstract data. We will use accuracy as a metric to test our model in experimental data.

Accuracy = Number of correctly classified items / All classified items.

Also, during model training, we compare prediction data with validation data and calculate the loss of Multi class logs to find the most efficient model. Log loss takes account of the uncertainty of the prediction based on how different it is from the original label and this will help to evaluate the model.

Exploring Data

For this project, the input format should be of the image type, because we want to insert the image and identify the dog type. The data set contains images of dogs and humans.

Dog Image Data: The set of dog data consists of 8351 train-filtered images (6,680 images), trial (836 photos) and valid directions (835). Each of these directories (train, test, active) has 133 folders corresponding to dog breeds. Images are of different sizes and backgrounds some images are not full-size. The details are not the same because the number of photos provided per genre varies. Few have 4 pictures and some have 8 pictures.

Human Image data set: Human data set contains 13233 human photos sorted by names (5750 folders). All images are 250 x 250 pixels. Photos have different background and different angles. Details are not the same because we have 1 photo for some people and lots of photos for others.

Sample Images from the dataset

Algorithms and techniques:

By performing this multiclass separation, we can use Convolutional Neural Network to troubleshoot. The Convolutional Neural Network (CNN) is an in-depth learning algorithm that can capture input images, assign value (readable bits and selections) to various objects / objects in an image and be able to distinguish one from another. The solution involves three steps. First of all, to get human photos, we can use an existing algorithm such as OpenCV's Haar Phase implementation which is supported by large. Secondly, to find dog photos we will use the VGG16 model that has been developed. Finally, after the image has been identified as a dog / human, we can transfer the image to a CNN model that will use the image and predict the best possible color for the 133 species.

Benchmark

The CNN model created from scratch must have accuracy of at least 10%. This can confirm that the model is working because a random guess will provide a correct answer roughly 1 in 133 times, which corresponds to an accuracy of less than 1%.

Data Pre-processing

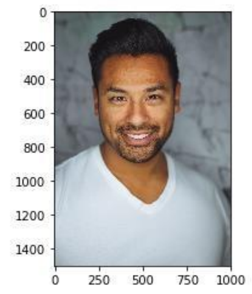
All the images are resized to 224*224, then normalization is applied to all images (train, valid and test datasets). For the training data, Image augmentation is done to reduce overfitting. The train data images are randomly rotated and random horizontal flip is applied. Finally, all the images are converted into tensor before passing into the model.

Implementation

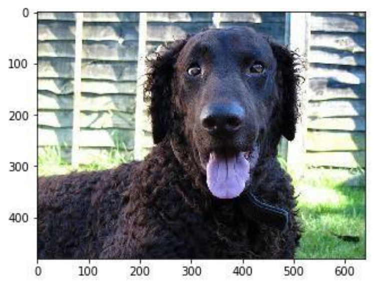
I have built a CNN model from scratch to solve the problem. The model has 3 convolutional layers. All convolutional layers have kernel size of 3 and stride 1. The first conv layer (conv1) takes the 224*224 input image and the final conv layer (conv3) produces an output size of 128. ReLU activation function is used here. 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 0.30 is added to avoid over overfitting.

Refinement

The CNN created from scratch have accuracy of 10%, Though it meets the benchmarking, the model can be significantly improved by using transfer learning. To create CNN with transfer learning, I have selected the Resnet101 architecture which is pre-trained on ImageNet dataset, the architecture is 101 layers deep. The last convolutional output of Resnet101 is fed as input to our model. We only need to add a fully connected layer to produce 133-dimensional output (one for each dog category). The model performed extremely well when compared to CNN from scratch. With 15 epochs, the model got 69% accuracy.



```
detected human resembles dog breed  German pinscher
```



```
detected dog breed Curly-coated retriever
```

```
detected dog breed Golden retriever
```

Sample outputs predicted using the model

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. 100% of dog faces were detected in first 100 images of dog dataset and 1% of dog faces 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 15 epochs, and the final model produced an accuracy of 69% on test data. The model correctly predicted breeds for 583 images out of 836 total images.

Accuracy on test data: 69% (583/836)

Justification

I think the model performance is better than expected. The model created using transfer learning have an accuracy of 69% compared to the CNN model created from scratch which had only 10% accuracy.

Improvement

The model can be improved by adding more training and test data, currently the model is created using only 133 breeds of dog. Also, by performing more image augmentation, we can avoid overfitting and improve the accuracy. I have tried only with ResNet 101 architecture for feature extraction, May be the model can be improved using different architecture.