Capstone Project **Oumou SANGARE**

Data Science Nanodegree January 31st, 2022

Dog Breed Classifier using CNN

Project Overview

This project is to use Convolutional Neural Networks to Identify Dog Breeds. Dog breed classification is really challenging due to the minimal inter-class variations. A Convolutional Neural Network (CNN) model can be used for this task. The final project will be able to identify the breed of a dog or if given an image of a human it will identify the breed which is similar to the human.

Problem Statement

The aim of the project is to build a pipeline to process real-world, user-supplied images. A CNN will be used to predict the breed of a dog from the input image. Given an image of a dog, the model will identify an estimate of the canine's breed. If supplied an image of a human, it will identify the resembling dog breed. The workflow for this project can be described as follows:

1. Import Datasets
2. Detect Humans using OpenCV's Haar feature-based cascade classifiers.
3. Detect Dogs using a pre-trained VGG-16 model.
4. Create a CNN to Classify Dog Breeds.
5. Write an Algorithm to return the breed if a dog is detected or the resembling breed if a human is detected.
6. Test the Algorithm.

Metrics

Accuracy will be used to evaluate both the solution model and benchmark model. Accuracy will be calculated with the help of the confusion matrix.

$Accuracy$ = ($TP + TN$) / ($TP + TN + FP + FN$)

TP: True Positive, TN: True Negative, FP: False Positive, FN: False Negative

Analysis

Data Exploration

The Datasets provided was properly labelled and already split into train, validation and test sets. The dogImages dataset consists of 8351 images of 133 breeds, 8 images from each breed. The human images were from the Labeled Faces in the Wild (LFW) dataset. It consisted of 13233 images of humans labelled according to their names. The data from both datasets are evenly distributed.

Algorithms and techniques:

CNNs are a kind of deep learning model that can learn to do things like image classification and object recognition. They keep track of spatial information and learn to extract features like the edges of objects in something called a convolutional layer. The convolutional layer is produced by applying a series of many different image filters called convolutional kernels to the input image.
Other than the CNNs, OpenCV's implementation of Haar feature cascade classifier was used to perform face detection of the human images.
On tasks involving computer vision CNNs outperforms other Neural Networks like MLP by a far.
This project requires two CNNs (model from scratch and model from transfer learning) mainly for classifying the breed and one for detecting dogs.

For detecting dogs, a pre-trained VGG-16 model was used, we need only check if the pre-trained model predicts an index between 151 and 268 (inclusive).
Transfer learning is a machine learning technique where a model trained on one task is re-purposed on a second related task.

Benchmark

The implementation of the model from scratch had a test accuracy of 1.0766%. My implementation of the model from transfer learning had a significant improvement from the model from scratch and it had a test accuracy of 81.4593%.

Methodology

Data Preprocessing

The images were resized and cropped to 224x224 pixels to achieve an input tensor of shape 224x224 which suits the pre-trained model and the model from scratch.
Images for the training set had undergone data augmentation to prevent overfitting and to get more generalized training data. Augmentation was done by randomly rotating, flipping, resizing and cropping the data. The validation and testing data was exempted from augmentation, but they were resized and cropped to 224x224 like the training set.
After resizing and cropping, the image data were converted to tensors and they were normalized.

Implementation

The implementation of this project involved 2 major parts: the CNN from scratch, the CNN from transfer learning.
The code snippet below shows the architecture of the CNN from scratch.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 223, 223, 8)       104
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 111, 111, 8)       0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 110, 110, 16)      528
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 55, 55, 16)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 54, 54, 32)        2080
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 27, 27, 32)        0
_________________________________________________________________
global_average_pooling2d_1 ( (None, 32)                0
_________________________________________________________________
dense_1 (Dense)              (None, 133)               4389
=================================================================
Total params: 7,101
Trainable params: 7,101
Non-trainable params: 0
_________________________________________________________________
```

It consists of 3 convolutional layers, after each convolutional layer is a Max Pooling layer. The fully connected layers use a Rectified Linear Unit (ReLU) activation function. The final fully connected layer has an output shape of 133 corresponding to the number of breeds.

Results

Model from scratch:
● Testing accuracy: 1.0766%
Model from transfer learning:
● Testing accuracy: 81.4593%

Improvement

- More variety is needed concerning the type of breeds the model predicts for humans.
- The model needs to pick up subtle differences between similar breeds. Some of them have similar colors and shapes but differ in size and other features.
- The model could also be improved on its ability to classify pictures with noise.