The Capstone Final Project Report

of the

Udacity Machine Learning Engineer Nanodegree

Funmilayo Olaiya

April 26, 2021

Project Overview

Domain background

The identification of different breeds of dog is very essential, most especially in terms of knowing and understanding the qualities of different dogs with their uniqueness, behaviour, health conditions and so many more.
This project made use of Convolutional Neural Network (CNN) pre-trained models from ImageNet called **VGG-16** to create a CNN model from scratch and **ResNet-50** to train the final model architecture to predict and identify dog breeds from images.

Problem statement

The goal of this project is to build a CNN classifier that would take the input of some images and return the expected output(s) in which these edge cases has to be fulfilled.

Case I: An image of a dog

If the image data supplied is that of a dog, the algorithm returns the exact breed of the dog successfully.

Case II: An image of a human

If the image data supplied is that of a human, the algorithm returns the closest resembling breed of a dog successfully.

Case III: An image is neither that of a dog nor human

If the image data supplied is neither a human nor a dog, then the algorithm returns an error.

Metrics

The metric used in measuring how the project is performing is through ensuring that it has an output **accuracy** of nothing less than 60%. The CNN classifier that was developed achieved an 81% accuracy which means that the model can identify a dog breed around 8 times out of 10 correctly.
Keeping in mind, the accuracy can always be measured with the following formula:

$$\textbf{accuracy(\%)} = \frac{\textit{total number of accurate predictions}}{\textit{total number of images in the provided dataset}}$$

A lower log loss metric depicts better predictions, and to further improve accuracy, we also have to consider the **log loss**, which is the most important classification metric based on probabilities such that we calculate the “log loss” to get the best working model that can provide accurate predictions.

Data Exploration and Visualisation

Two datasets have been provided by Udacity in which they both contain **human images** and **dog images**.

The Human Images Dataset:

1. The total number of human images in this particular dataset is 13,233. The human dataset comprises 5749 folders, each containing human pictures. The structure of this dataset is very different to that of the dog dataset as it is not split into "test", "train" and "validation" folders.
2. This dataset is also highly imbalanced because a person might have only one image while other persons might have more than one image.
   Figure 3 below is a histogram plot depicting the differences.

Figure 1: Sample human image from the human images dataset.

Figure 2: Sample dog image from the dog images dataset.

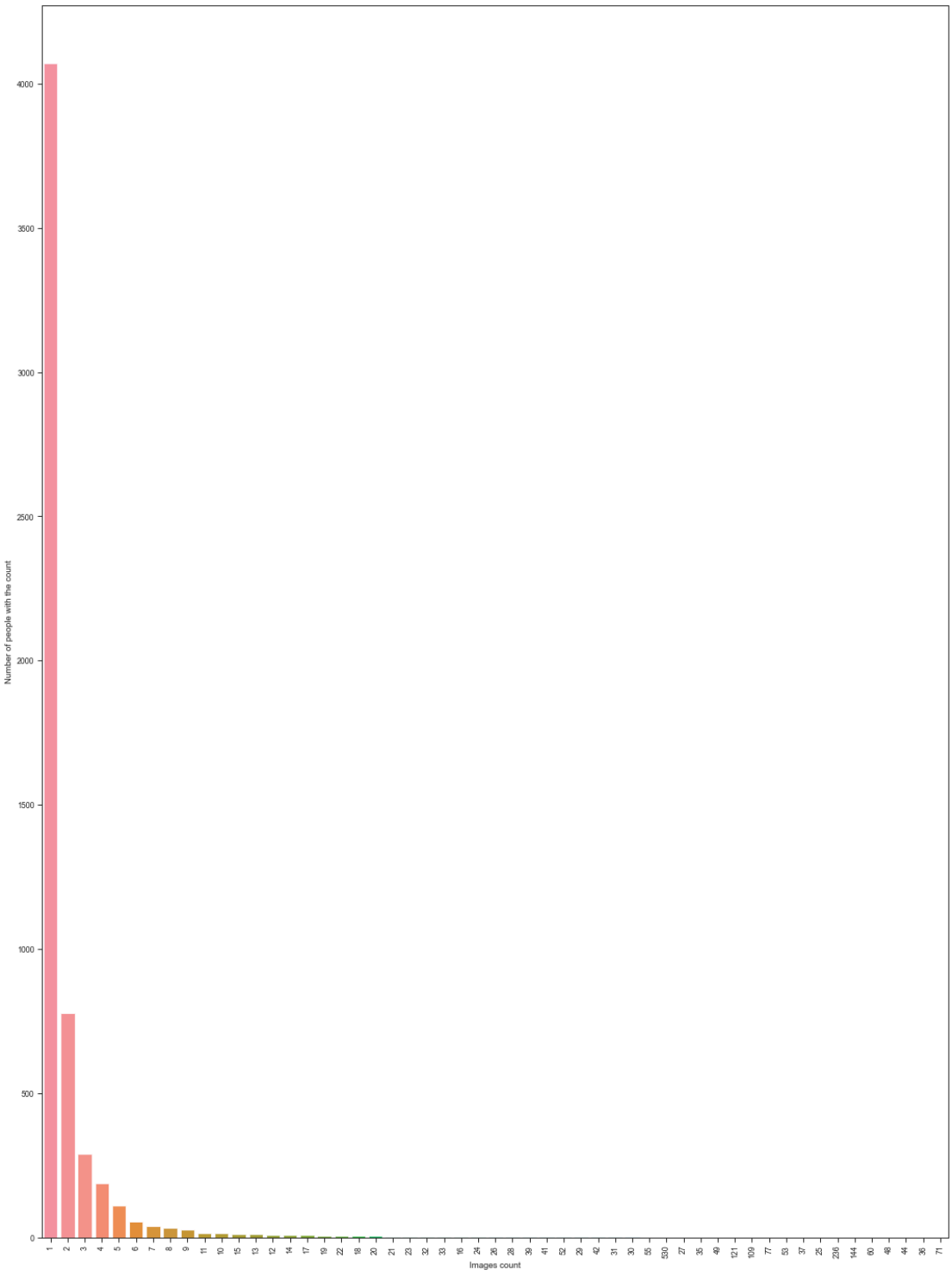

Figure 3: A plot showing the count of images for different persons in the human images dataset. On the (y) axis shows the "number of people with the count" and on the (x) axis shows the "images count."

The Dog Images Dataset:

The total number of dog images in this particular dataset is 8,351 in which are structure in different folders such as:

I. The train folder - which comprises 6,680 dog images.
II. The test folder - which comprises 836 dog images.
III. The validation folder - which comprises 835 dog images.

Each of these folders in the dog images dataset has 133 folders which correspond to the 133 dog breed outputs.

By analysing the dog images dataset further, the number of images in each folder in the **train** dog images dataset varies, which makes the dataset imbalanced.

Figure 4 below is a histogram plot depicting the differences.

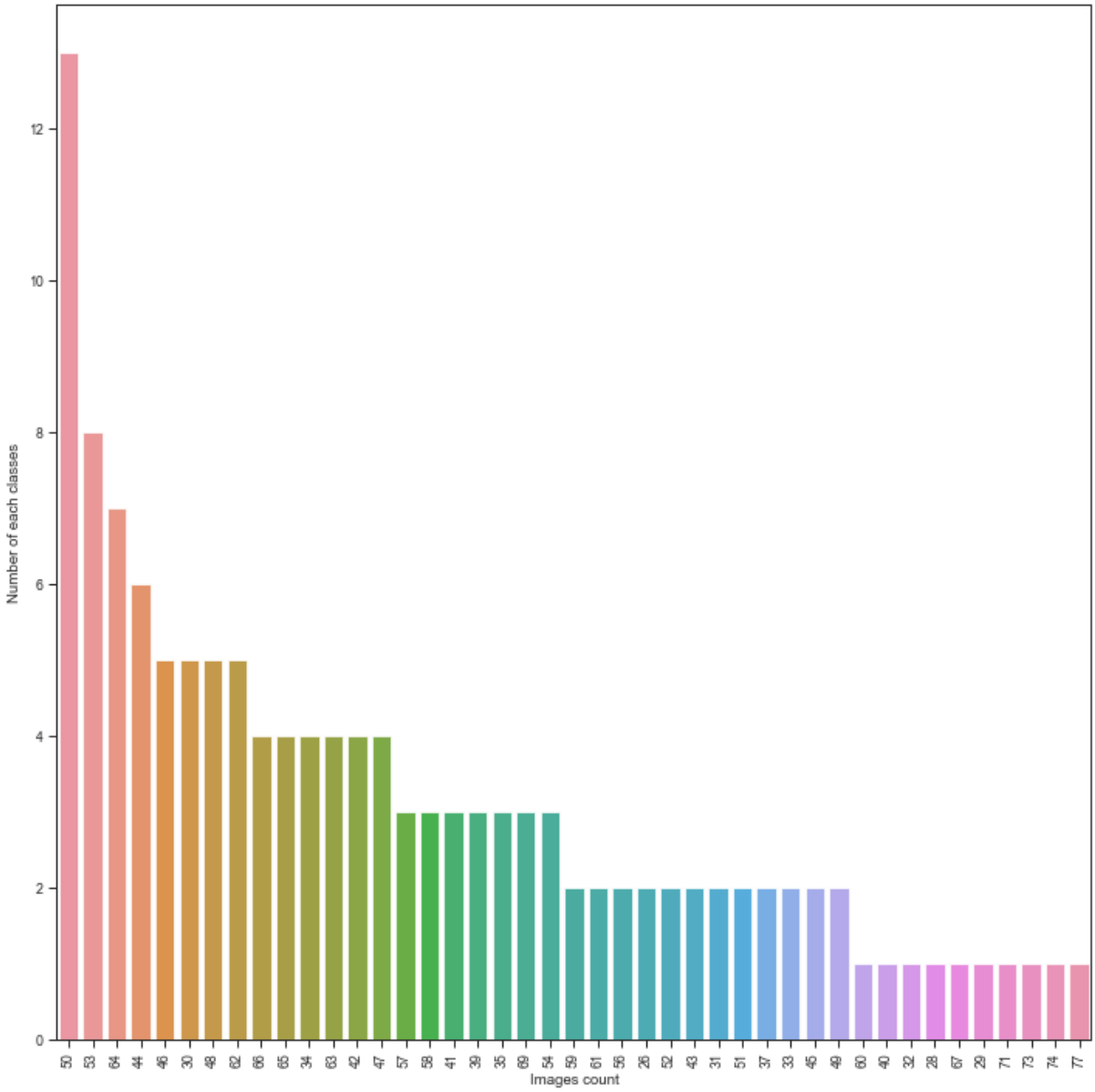

Figure 4: A plot showing the count of images in different classes in the train dog images dataset.

Algorithms and Techniques

In this section, we discuss the techniques and steps used in achieving the creation of the CNN classifier.

Step 1: Human detection:

The OpenCV's implementation of **Haar feature-based cascade classifier** is used in the detection of human faces in images. This is a machine learning-based approach in which a cascade function is trained from a lot of positive and negative images and later used to detect objects in other images.

Step 2: Dog breed detection:

A pre-trained model called the **VGG-16** model, which has been trained on the ImageNet dataset is used. We input the images into the pre-trained model and get a predicted output.

Using this model, a CNN was created from scratch and before the images were fed into the model, some data augmentations had to take place like; resizing, cropping, slight rotations and horizontal flips, just to expand the training set and prevent overfitting and we achieved an accuracy of 11%.

To achieve our final CNN architecture, we made use of a pre-trained model called, **ResNet-50** which is a Convolutional Neural Network that is 50 layers deep, and that has been trained on more than a million images from the ImageNet database. Finally, high accuracy of 81% was achieved.

Benchmark

1. The CNN created from scratch must achieve an accuracy of nothing less than 10%.

2. The CNN created by making use of a transfer learning model must achieve an accuracy of nothing less than 60%, which depicts that at least, more than half of the dogs should be correctly classified into their breeds.

Data Preprocessing

1. For the human face detector, not much preprocessing was done except to convert the colour images to grayscale. This step is one of the standard steps taken in image processing/face detection because monochrome(grayscale) images are very sufficient and less information is desired for each pixel, also which makes it easier to process than colour images.

2. For the dog breed detector, there were some preprocessing done before feeding the images to the model. The images were resized generally to 256px, cropped to sizes 224X224 to the centre, and they are also normalised using a mean of 0.5 each. The images are also flipped and rotated randomly.

Implementation

After the needed datasets are obtained and downloaded, **OpenCV's implementation of Haar feature-based cascade classifier** is implemented to detect human faces in images.
The pre-trained model, **VGG-16** was used in creating the CNN classifier from scratch and the **Resnet-50** pre-trained model was also used to create the final classifier.
Finally, an algorithm was written to classify the images such that:
I. If the image is that of a human, it should return the closest resembling dog breed.
II. If the image is that of a dog, it should return the accurate breed of the dog.
III. If the image is neither a dog nor a human, it should return an error statement.

Refinement

I tried training the CNN classifier created from scratch using the **VGG-16** pre-trained model with only 5 epochs and with a learning rate of 0.05 and it brought about only a 9% accuracy. I had to increase the epochs to 10 to increase the output accuracy. Finally, I got a 11% accuracy ,which significantly means that if the number of epochs increases, a higher percentage accuracy can be achieved.

For the CNN classifier created with the pre-trained model, **ResNet-50**, with a learning rate of 0.001 and only 1 epoch, after training, it achieved an accuracy of 57%. Later the epochs were increased to five and it achieved an accuracy of 81%. This means that if we increased the number of epochs to 10 or 20, the accuracy can go higher in percentage and log loss minimised.

Model Evaluation and Validation

The models were validated against the validation set during the training of the models.

For the model created from scratch, with a learning rate of 0.05 and only 10 epochs, it achieved higher accuracy than 10%, that which is 11%.

For the final model created using Transfer Learning, it has an accuracy of 81% and used a learning rate of 0.001 during the training, this shows that it predicted at least 677 testing images accurately. The learning rate of 0.001 delivered good results such that only 5 epochs supplied were enough to achieve a decent accuracy.

Justification

The final model has an accuracy of 81% which depicts a better performance than the required accuracy of 60%. With the various information provided in other sections of this report, I believe an 81% accuracy is reasonable enough to be used in an application and which can be consistently improved over time.

Areas I believe can be improved on are; the ability to be able to detect the breeds of more than 1 dog in an image, the model might have better accuracy when approached with a different architecture other than **ResNet-50**, also we could have more than **133** classified breeds of dogs(more training and test data).

References

1. Punyanuch, B., Worapan K., Sarattha K., Kittikhun T. (2020). "Knowing Your Dog Breed: Identifying a Dog Breed with Deep Learning", International *Journal of Automation and Computing.* 18, pp. 1.

2. Maanav Shah, DogBreed Classifier using CNN [Blog post]. https://medium.com/@maanavshah/dog-breed-classifier-using-cnn-f480612ac27a

3. Dan Becker, What is Log Loss? [Blog Post]. https://www.kaggle.com/dansbecker/what-is-log-loss

4. Cascade Classifier, [Documentation]. https://docs.opencv.org/master/db/d28/tutorial_cascade_classifier.html

5. ResNet-50, [Documentation]. https://www.mathworks.com/help/deeplearning/ref/resnet50.html;jsessionid=fefffdac2f52dd6c07cdf9ffe29f#:~:text=ResNet%2D50%20is%20a%20convolutional,%2C%20pencil%2C%20and%20many%20animals.

6. Grayscale Images, [Documentation]. https://homepages.inf.ed.ac.uk/rbf/HIPR2/gryimage.htm