CAPSTONE PROPOSAL

DOG BREED CLASSIFIER WITH CNNs

Paulo Victor de Sousa Moura April/2021

DOMAIN BACKGROUND

Humans interact with the world in many ways. One of the most important of these interactions is through vision. Humans can easily interpret the three-dimensional world and analyze and identify objects seen. Today's modern computers are capable of working with captured images, and the increasing use of the Internet in recent decades has provided us with gigantic amounts of image data. As a result, a lot of development is taking place to make a machine extract meaning from the images.

The field of image processing is one of the subfields of machine and deep learning, along with signal processing and many others and is an active area of research. This field is based on image processing by means of mathematical algorithms and uses these algorithms to fulfill many objectives, such as classification, forecasting and even image generation, such as the use of Adversarial Neural Networks (GANs). Some real-world examples include optical character recognition, medical images, self-driving cars, face recognition, object classification, etc. With the advancement in computer hardware, we have been able to perform increasingly expensive tasks, which are one of the reasons why deep learning has taken off. An innovative application of deep neural networks that emerged in the 2012 ImageNet challenge was the use of “Convolutional neural networks” to win the ImageNet competition.

A convolutional neural network, CNN for short, is a specialized type of neural network for processing data that has the structure of a grid. Classification of images with CNN came to light in 1994 with LeNet5 by Yann LeCun.

Given that people can be helped by an image classifier that recognizes dog breeds, in this project, I am interested in seeing how convolutional neural networks work to recognize dog breeds using a dog breed classifier.

PROBLEM STATEMENT

How can we use convolutional neural networks to recognize the dog breed classifier?

One of the main challenges in the field of image processing is to allow a system to recognize the appearance of things and what are the differences between them. The aim of this project is to build an algorithm that will be able to distinguish between dog breeds and overcome the challenge that many dog breeds would be very similar to. Based on a dog's image, an algorithm should provide an estimate of the dog's breed. If an image of a person is provided, the algorithm should reproduce the most similar breed of dog.

Therefore, we face the following main problems in this project.

1. Create a model to detect a human
2. Create a model to predict the breed of dog
3. Create an algorithm, which will return the expected breed of dog for the image.
4. Achieve 60 percent or more accuracy in the breed classification

DATASET AND INPUTS

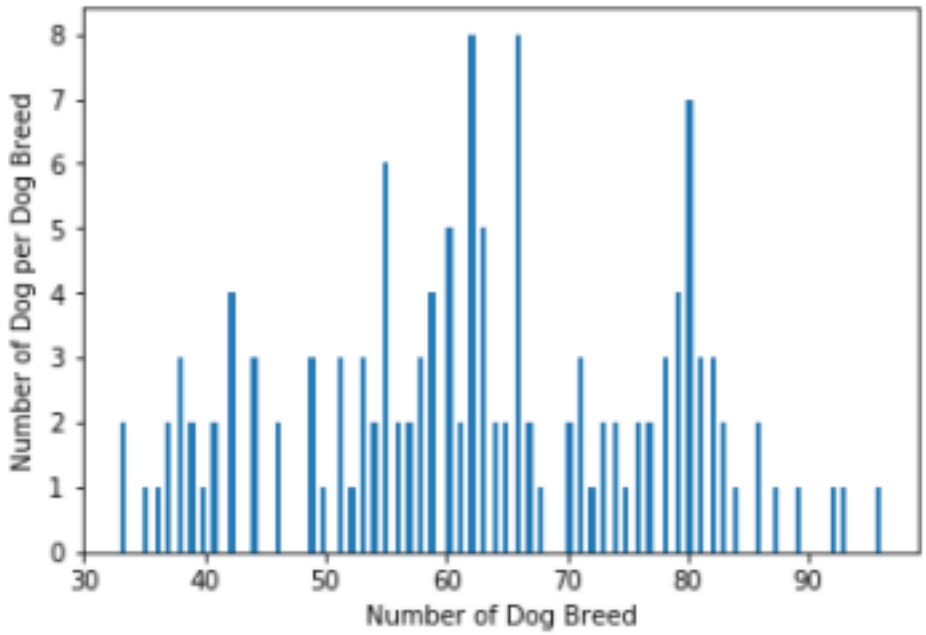
The dataset to be used in this project is provided by Udacity, which contains images of humans and dogs as well. This dataset is already separated into a training, testing and validation folder.

To classify dog breeds, we will use a dataset that contains 8,351 images of dogs. Therefore, we have 6,680 images to train the model, 836 images for testing and 835 images for validation, which means that 80% of the total data was kept for training and 10% each for validation and testing. The entire data set contains 133 different breeds.

To detect humans, we will use the LFW (Labeled Face in the Wild) dataset from the University of Massachusetts. The LFW dataset consists of face photos. Each face was tagged with the person's name. This data set contains 5,750 common images of more or less ordinary people. These images are used to test the performance of the Human Face Detector.

SOLUTION STATEMENT

The solution is to design a CNN model that is able to estimate as much as possible the breed of a dog that is present in a photo. To do this, one must recognize beforehand whether a person or a dog is present in the image. If a person is present, the similar breed of dog is identified. If a dog is present, an estimate of the breed of the dog is provided.



From the figure we can realize that the data set is imbalanced, it does not have a similar amount of data for each of the classes. The number of images in each class varies from 26 images to 77 images, mean of 50 images per breed. This imbalance can also be seen in the test and validation sets.

First, we need to find a way to detect human faces. For this problem, OpenCV provides the classifier Haar Cascade. The classifier is trained from images with a machine learning approach based on a cascade function, to detect objects in images.

To classify whether the image is of a dog, we realize that the ImageNet dataset has several labeled dog images. Thus, it would be a good approach to directly use one of the pre-trained models from the ImageNet competition to find out if it is a dog. Finally, to predict the exact breed of the dog, we can again use one of the models from the ImageNet competition. This time, we will use only the pre-trained weights of the convolutional layer and retrain the fully connected layers for our use case of 133 classes of dog breeds.

BENCHMARK MODEL

We are building two models in two different ways:

● Building a CNN from scratch. That should exceed 10% accuracy. This is a challenging task due to the fact that image processing models need many iterations and times that take a long time to finish training

● Using a pre-trained model. It must achieve an accuracy of at least 60%, so that it can be used successfully in a dog breed classifier application.

EVALUATION METRICS

The objective here is to compare the performance of my model with that of the reference model. Therefore, the percentage of accuracy would be an appropriate metric for judging the performance of our model, as it provides a solid measure of how well our model is performing, limited between 0 and 100. Also, because the benchmark model specifies only the precision.

$$accuracy\ \% = \frac{number\ of\ correctly\ predicted\ images}{total\ number\ of\ images} \times 100$$

Additionally, due to unbalanced data set, we can evaluate performance using the F1-Score metric, which is calculated from Precision and Recall metrics.

$$F1Score = 2 * Precision * \frac{Recall}{Precision + Recall}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

Where TP stands for the number of true positive, FP stands for the number of false positives, FN stands for the number of false negatives and FP stands for the number of false positive.

PROJECT DESIGN

The design steps for this project are as follows:

1. Import data sets

The data described in section 3 'Datasets and inputs' are imported.

2. Detect humans

An algorithm capable of recognizing human faces using OpenCV and the integrated Haar Cascade will be developed.

3. Detect dogs

The pre-trained VGG-16 model, which comes with the weights that were trained in the ImageNet dataset, can be used here to design a dog detector.

4. Create a CNN to classify dog breeds from scratch

A CNN model from scratch will be developed to classify dogs. Data augmentation (rotating, resizing, cropping) from the original images of dogs will be used to train the model.