Udacity Machine Learning Engineer Nanodegree
Capstone Project

Dog Breed Classification Using Convolutional Neural Network

Ersin Genel

3 March 2021

Domain Background

According to the World Canine Organisation (Fédération Cynologique Internationale), there are over 300 recognised dog breeds in the world. In this project, we will try to determine the breed of a dog using an image. More interestingly, if an image of a human is provided instead, our algorithm will identify the resembling dog breed. We'll use CNN (Convolutional Neural Network) models for classification. In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery which perfectly fits our project.

Problem Statement

We need to build a machine learning model that uses images as an input. It should return an estimation for the breed of the dog if there is a dog in the image. If a human is detected in the image the algorithm should find the dog breed that most closely resembles. Our model can be consumed by a web application, web service or mobile application when ready. It can be helpful if the model can handle some cases such as user-supplied images without dog or human.

Datasets and Inputs

Our input will be images in this project as we can easily guess. There should be images of dogs and humans. Fortunately, all resources we need are provided by Udacity. We have two datasets:

Human Dataset: https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/lfw.zip There are 13233 human images in this dataset that will be used to detect human faces. We'll use Haar Cascade Classifier to detect human faces in images.

Dog Dataset: https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/dogImages.zip There are 8351 dog images in this dataset that will be used to detect dog's breeds. We'll use the VGG-16 model and ImageNet to detect dogs in images.

Solution Statement

After importing datasets, we need to detect humans and dogs. In order to detect human faces in images, we'll use Haar Cascade Classifier which is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. We'll use the VGG-16 model, along with weights that have been trained on ImageNet to be able to detect dogs. After completing identifying humans and dogs, we need to create a CNN to classify dog breeds that must obtain a test accuracy of at least 10%. Lastly we'll create another CNN using transfer learning to validate and test the results. Our model must attain at least 60% accuracy on the test set.

Benchmark Model

Our CNN model without transfer learning must obtain a test accuracy of at least 10%, the CNN model using transfer learning that we'll create finally, must obtain at least 60% accuracy as a benchmark model provided by Udacity.

Evaluation Metrics

In order to evaluate our machine learning algorithm we'll use Classification Accuracy that is the ratio of number of correct predictions to the total number of input samples. We can try out other evaluation metrics if needed.

$$Accuracy = \frac{TrueNegatives + TruePositive}{TruePositive + FalsePositive + TrueNegative + FalseNegative}$$

Project Design

We'll follow the steps below to create our model:

Step 0: We'll import dog and human datasets from the provided sources.
Step 1: In this step we'll detect humans using Haar Cascade Classifier.
Step 2: Next step will be to detect dogs, we'll use the VGG-16 model and ImageNet to detect dogs in images.
Step 3: After completing identifying humans and dogs, we need to create a CNN to classify dog breeds that must obtain a test accuracy of at least 10%.

Step 4: Then we'll create another CNN using transfer learning to dog breeds. We need to validate if it obtains at least 60% accuracy.
Step 5: We'll validate and test our algorithm.

References

https://en.wikipedia.org/wiki/Dog_breed
http://www.fci.be/en/Presentation-of-our-organisation-4.html
https://en.wikipedia.org/wiki/Convolutional_neural_network
https://docs.opencv.org/master/db/d28/tutorial_cascade_classifier.html
https://developers.google.com/machine-learning/crash-course/classification/accuracy
https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234
https://medium.com/@erika.dauria/accuracy-recall-precision-80a5b6cbd28d