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. There are similar problems that solved by deep learning based models in the following links https://besjournals.onlinelibrary.wiley.com/doi/10.1111/1365-2656.12780 (A computer vision for animal ecology), https://www.pnas.org/content/115/25/E5716 (Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning) and https://atrium.lib.uoguelph.ca/xmlui/handle/10214/18056 (Deep Learning Based Computer Vision for Animal Re-Identification).

Problem Statement

We need to build a machine learning model that uses images as 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 a 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 have 5749 folders that contain different counts and one dimension (250x250 px) of human faces. There are different colours and backgrounds in the images. Human images are imbalanced.
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 have 133 folders that contain different counts and dimensions of a dog breed. There are different colours and backgrounds in the images, and also two or more dogs in an image. Dog images are imbalanced.

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 Multi Class Log Loss to handle class imbalances. As I see the same evaluation metrics used for a similar Kaggle competition which can be very helpful to compare: https://www.kaggle.com/c/dog-breed-identification/overview/evaluation

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 use an OpenCV model to detect human faces. Haar feature-based cascade classifier is advised by Udacity to solve this problem. Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.
Step 2: Next step will be to detect dogs, we'll use the VGG-16 model and ImageNet to detect dogs in images. VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition''. The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous models submitted to ILSVRC-2014. It improves AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another. VGG16 was trained for weeks and was using NVIDIA Titan Black GPU.
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%. The train, test and validation data is provided in the sources, we'll verify the model using those data.
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 write our own algorithm that accepts an image file path and identifies whether the provided image contains a human or dog. If a dog is detected in the image, our algorithm should return the predicted breed. If a human is detected in the image, it should return the resembling dog breed. If neither is detected in the image, it should return an error. As a final step, we'll test and validate our own 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://neurohive.io/en/popular-networks/vgg16/