Tuesday, May 5, 2020

Elias Junior Ghantous
Udacity Machine Learning Engineer Nanodegree
Capstone Project

Dog Breed Classifier Using Convolutional Neural Networks

Domain Background

Classifying dog breeds has long been a popular problem in the machine learning industry. In fact, "man's best friend" has countless Kaggle datasets and submissions dedicated to them pertaining to a multitude of questions one might ask about dogs. Ultimately, the question of "what breed is that dog?" is one I chose to answer, by suggestion of the Udacity Machine Learning Nanodegree course. My goal is to develop a machine learning model to answer the question, taking the guesswork out of the situations people face when they fail to recognize the breed of dogs passing them by. A final report will be produced detailing the methodology and results with hopes that a web app can be built utilizing this model.

Problem Statement

The goal is to build a machine learning model which could be deployed via a web app, helping users predict and classify dog breeds. This is a multiclass classification problem where supervised learning can be used. The final algorithm must perform two tasks:

- ***Detection of human faces:*** If supplied an image of a human, the model will identify the resembling dog breed, similar to that of popular Buzzfeed questionnaires e.g. "Which type of garlic bread are you?"
- ***Detection of dog faces:*** If supplied an image of a dog, the model will identify the dog breed.

Datasets and Inputs

The dataset for this project which includes images of both dogs and humans is provided by Udacity.

'dogImages' dataset: The dog image dataset has 8351 total images which are sorted into train (6,680 Images), test (836 Images) and valid (835 Images) directories. Each directory has within it 133 folders corresponding to dog breeds. The images are of different sizes and different backgrounds with some of the images being of smaller sizes than the others. The number of images per breed varies as well, an issue which could possibly affect the accuracy of the model produced.

'lfw' dataset: The human dataset contains 13233 total human images which are sorted in alphabetical order of the names of the humans (5749 folders). All images are of size 250x250. Images have different background and different angles. The number of images per person varies, making the dataset unbalanced and potentially affecting the model accuracy.

Solution Statement

In order to perform the multiclass classification, I am going to use a convolutional neural network. In deep learning, a **convolutional neural network** (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics. The network applies weights to specific objects in each image in order to 'see' the image differently to others.
First, we must be able to detect human images. To do this, I will use an existing algorithm, OpenCV's implementation of Haar feature based cascade classifiers. Second, in order to detect dog images, I will use a pretrained VGG16 model.
Finally, after identifying the image as either dog or human, we pass this image to a CNN which will process the image and predict the breed of dog out of all possible 133 breeds.

Benchmark Model

The CNN model created from scratch must have accuracy of at least 10%. This would confirm that the model is working, ruling out the chance of it guessing randomly which would return a correct answer in roughly 1/133 times (<1% accuracy).
The CNN model created using transfer learning must have accuracy of 60% or higher.

Evaluation Metrics

I am going to use multiclass log loss as the evaluation metric. Due to the varying number of pictures for each breed in the dataset, accuracy would not be suitable. The metric log loss is such that it takes into account the uncertainty of the prediction made relative to the actual label and so the model would be evaluated fairly.

Project Design

Step 1: Import the necessary dataset and libraries. Pre-process the data and create the train, test and validation datasets. Perform image augmentation on training data.

Step 2: Detect human faces using OpenCV's implementation of Haar feature based cascade classifiers.

Step 3: Create a dog detector using the pretrained VGG16 model.

Step 4: Create a CNN to classify dog breeds from scratch. Train, validate and test the model.

Step 5: Create a CNN to classify dog breeds using Transfer Learning with resnet101 architecture. Train and test the model.

Step 6: Write an algorithm to combine the dog detector and human detector.

a) If a dog is detected in the image, return the predicted breed.
b) If a human is detected in the image, return the resembling dog breed.
c) If neither is detected, provide output that indicates the error.

References

Original repo for Project - GitHub: *https://github.com/udacity/deep-learning-v2pytorch/blob/master/project-dog-classification/*

Resnet101:
https://pytorch.org/docs/stable/_modules/torchvision/models/resnet.html#resnet101

Pytorch Documentation: *https://pytorch.org/docs/master/*

ImageNet training in Pytorch:
https://github.com/pytorch/examples/blob/97304e232807082c2e7b54c597615dc0ad8f6173/imagenet/main.py#L197-L198

Log loss: *http://wiki.fast.ai/index.php/Log_Loss*

CNN definition: *https://en.wikipedia.org/wiki/Convolutional_neural_network*