Udacity Machine Learning Engineer Nanodegree

CapStone Project Proposal

DOG BREED CLASSIFIER

Terrence Goh

Apr 14, 2022

DOMAIN BACKGROUND

When computer vision started in the 1960s, its aim was to try and mimic human vision systems and ask computers to tell us what they see, automating the process of image analysis. Early researchers were extremely optimistic about the future of this exciting field. This kind of technology is widely regarded as the precursor to artificial intelligence and potentially could transform the world.

Early research attempts met with limited success, and with computer hardware and computing power at that time, it was not able to solve even the basic tasks.

After more than 50 years had passed, the world now has the Internet with a large collection of data and photos uploaded by everyday users. Computing power has grown tremendously and hardware costs keep going down every year.

The growth of the internet and small, more powerful computers are a big catalyst for a team from the University of Toronto to win an image classification competition in 2012 at the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). They created an artificial deep neural network named AlexNet that won the competition with an error rate of 15.3%, versus the runner up 26.2% [1].

Nowadays, the gold standard for image recognition tasks is a class of deep neural networks called ***convolutional neural networks*** (***CNN***s for short) and this project will explore using CNNs to solve an image classification problem.

PROBLEM STATEMENT

The goal of this project is to develop a machine learning algorithm that takes a photo or an image as an input, and classify it according to the conditions below:

1. Given an image of a dog, identify an estimate of the canine's breed.
2. Given an image of a human face, identify the resembling dog breed.

DATASET AND INPUTS

The dataset is provided by Udacity which consists of thousands of images, both humans and dogs. The photos are sorted into three folders: test, train and validation. They are used for each step in the machine learning(ML) workflow.

SOLUTION STATEMENT

Referring to the problem statement, the algorithm needs to identify if the photo is a human or a dog. If it is a dog, the algorithm will find the closest match of the dog breed. We can use one of the models trained from the ImageNet dataset, which has a large number of images labelled with dog. Some model candidates are Alexnet, VGG16, Inception and Resnet50, just to name a few.

If the photo is a human, the algorithm will try to find the dog breed that closely resembles the human face…just for fun. However, to detect a human face in the first place, we will need another algorithm more suited for this approach, which is Haar Cascades[2] from the OpenCV software library.

Back to identifying the dog breed, the pre-trained models above can only identify if the image contains a dog, but not classifying the dog breed itself. We have to apply transfer learning[3] and customize the models to our needs.

BENCHMARK MODEL

The benchmark model(s) for this project are developed with the performance metrics given below.

CNN model developed from scratch

Attain a test accuracy of ***at least 10%.***

Since our model is developed from scratch with little model-building experience and limited model training time, achieving 10% is relatively reasonable. After this, we will apply transfer learning on what we have learnt on pre-trained CNN models to achieve higher accuracy.

Pre-trained CNN model

Attain a test accuracy of ***at least 60%***.

Using transfer learning, we will transfer the model weights from the CNN developed from scratch, apply on a pre-trained model and re-run the training step. It should be good enough for our “dog breed classifier” application.

EVALUATION METRICS

The most common evaluation metric for image classification task is accuracy, which is the fraction of correct predictions over total number of predictions. The equation is expressed as a percentage as shown below:

$$accuracy\,\% \;= \frac{correct\ predictions}{number\ of\ predictions} \;\times\; 100$$

PROJECT DESIGN

This section outlines the steps in a little more detail on how to approach the problem. Do note that it might change a little as we gain more knowledge in the model training and evaluation phase of the ML workflow.

Import Data

The dataset provided is imported and explored to check the quality of the data, whether the photos are clear or obscured, or is it “clean” (non-dog photo in the dataset). It is also good practice to check if the number of photos are sufficient (at least 10 or more) and whether the samples per dog breed category is equally balanced to avoid overfitting to a particular category.

Human Face Detection

Once the dataset preparation is done, the Haar Cascade classifier is applied to find human faces. The image is converted to grayscale, fed into the input of the algorithm, detect the edge or line features in the image to determine if there is a human face or not.

Dog Detection

For this task, the pre-train model VGG16 model is used since it has already been trained on a very large ImageNet dataset (1000 categories, 1.2M images [4]) for many hours and fine-tuned to identify 1000 categories. If the image class predicted falls within a value of “151-268” (both inclusive), very likely the image is a dog since the dictionary keys fall under categories of different dog breeds, in the ImageNet dataset.

Dog Breed Classification

Once the step above is done, we can transfer the model parameters trained earlier to another deeper CNN model like ResNet with more hidden layers for better accuracy.

The first step is to perform image augmentation (flipping ,resizing, rotating, cropping) to provide a more generalized dataset of different images and avoid overfitting. Some photos could be a dog lying down and not a straight-up portrait photo, and we still want our model to be robust and classify the dog breed when given an input photo like these.

Second step is to transfer the model weights to ResNet and unfreeze the layers closer to the output. The fully connected layer will also be fine tuned to match the number of dog breed classes in our dataset. This will help the ResNet to re-learn classification from our data and model weights, and re-run the training loop.

Lastly, the pieces above are integrated into a final working demo of the project. Given an input dog image, the dog breed will be stated. If given a human photo, the dog breed that closely resembles the picture will be given.

If time permits, a stretch goal would be to have the model deployed on an AWS instance and have a static webpage to upload a photo and display the result of the identified dog breed.

References

[1]. Jerry Wei, July 3rd, 2019. AlexNet: The Architecture That Challenged CNNs.
https://towardsdatascience.com/alexnet-the-architecture-that-challenged-cnns-e406d5297951

[2]. OpenCV v4.0 Cascade Classifier.

https://docs.opencv.org/4.0.0/db/d28/tutorial_cascade_classifier.html

[3]. Jason Brownlee, Dec 20, 2017. A Gentle Introduction to Transfer Learning for Deep Learning.

https://machinelearningmastery.com/transfer-learning-for-deep-learning/

[4] Olga Russakovsky*, Jia Deng*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. (* = equal contribution) ImageNet Large Scale Visual Recognition Challenge. IJCV, 2015.

https://www.image-net.org/challenges/LSVRC/2014/