Machine Learning Engineer Nanodegree Capstone Proposal

Dog Breed Classifier

Si Yan

April 21, 2021

Domain Background

The main challenge of this project is to build an image classifier with sufficient accuracy. Dog breed classification is challenging for a few reasons:

1. Large numer of classes – more than 300 breeds have been recognized so far according to the Fédération Cynologique Internationale (FCI).[1] Large size of training data is needed to cover all classes.
2. Low inter-class variation – dogs in different breeds may be very similar in appearance.
3. High intra-class variation – one dog breed may have different colors and sizes.

For such a challenging image classification problem, Convolutional Neural Networks (CNN) is the best choice. CNN has been widely applied in facial recognition, text classification, self-driving cars, medical image processing, and more.[2] Unlike traditional multilayer perceptron (MLP) network that is a fully connected network, CNN has multiple convolution and pooling layers before the last fully connected layer. This architecture takes advantage of the hierarchical pattern in data and assemble patterns using smaller and simpler patterns. Therefore, CNN has high prediction accuracy and computational efficiency on classification problems.

In this project, I will use CNN to build the dog breed classifier for its high accuracy.

Problem Statement

The goal of this project is to build an algorithm that accepts a user-supplied image path as input and makes classification on the image. The algorithm first determines the input image contains a dog, human, or neither. Then, it output predictions based on the detected type:

If a dog is detected, identify an estimate of the canine's breed.

If a human is detected, identify the resembling dog breed.

If neither dog nor human is detected in the image, provide an output that indicates an error.

Datasets and Inputs

All of the human and dog images are provided by Udacity. The dataset has 13233 human images and 8351 dog images. The dog images will be used to train, validate and test CNN classifiers. The provided dog image dataset has pre-defined training, validation and test sets of images. Out of the total 8351 dog images, 80% (6680 images) are in the training set, 10% (835 images) are in the validation set, and 10% (836 images) are in the test set. There are 133 classes of dog breed in the dog dataset.

All images will be resized to 244 x 244 pixel before input into the model.

Solution Statement

A pre-trained Haar feature-based cascade classifiers from OpenCV is used for human detection, and a pre-trained VGG-16 model is used to detect dog in images.

To ensure a high classification accuracy with limited computation resource, transfer learning technique is used to build the classifier to identify dog breed from dog images and resembling dog breed from human images. The base classifier is the pre-trained ResNet-50 network, but the final fully-connected layer is replaced with a linear layer with 133 output classes.

Benchmark Model

The benchmark model is the CNN that I build from scratch, which has accuracy of 15% with the test images. This model has:

- 3 convolutional layers, each followed by a ReLU and a pooling layer.
- 2 dropout layers, with 25% drop-off rate
- 2 fully-connected layers

Evaluation Metrics

The percentage accuracy is used to evaluate the model performance. Accuracy is one of the most widely used metrics for classifier evaluations. This metric is calculated using the number of correct predictions divided by total number of predictions.

Project Design

Recommended by the project instruction, this project will be finished in seven steps.

Step 0: Import datasets for human and dog images.

Step 1: Implement the human face detector.

Step 2: Implement the dog detector.

Step 3: Create the benchmark model to classify dog breeds from scratch. See the details of this CNN in Benchmark Model section.

Step 4: Create a CNN to classify dog breeds using Transfer Learning. The base classifier is the pre-trained ResNet-50 network.

Step 5: Implement the algorithm that accepts a file path to an image and first determines whether the image contains a human, dog, or neither. Then,

if a dog is detected in the image, return the predicted breed.

if a human is detected in the image, return the resembling dog breed.

if neither is detected in the image, provide output that indicates an error.

Step 6: Test the algorithm implemented in Step 5 with some images.

Reference

1. http://www.fci.be/en/Nomenclature/

2. Bhandare, A., Bhide, M., Gokhale P., Chandavarkar R., Applications of Convolutional Neural Networks, IJCSIT, Vol. 7 (5) , 2016, 2206-221.