Proposal: Use Convolutional Neural Networks to Identify Dog Breeds

Fei Xia

July 2019

1 Domain background

Nowadays, classifying animals, especially pets, have been one of the major needs for pet owners and animal enthusiasts. For most of the cases of animal classification, the customers upload an image and expect a classification result to be returned. Convolutional Neural Networks (CNN) are commonly used for image related machine learning problem. Here, we will work with dog breed classification problem specifically and will work on building a CNN model for this problem.

2 Problem statement

Using machine learning model to aid computer vision is a very interesting and promising direction. In this project, we want to build a pipeline to process user-supplied images. Given an image of a dog, the algorithm will identify the breed of the dog. If there exists a human in the image, the code will return the resembling dog breed.

3 Datasets and inputs

The training and testing datasets are based on the images provided by Udacity. It is based on real-world dataset and have already been saved under the directory $*$/dog_images for dog images and $*$/lfw for human images. The images will be downloaded to local directory, unzipped and then be imported using glob and converted to narray with the dimensions $(N \times pixelsX \times pixelsY)$, where N is the number of the images (13233 in total for human images. 8351 in total for dog images.) and $pixelsX$, $pixelsY$ are the dimensions of each single image. For human face images, they have the same dimension: 250×250. For dog images, they have variable dimension: roughly around 400×400, which will be resized to 256×256. The distribution for the training set for each class is plotted below. As we can visualize clearly, the training dataset is unbalanced.

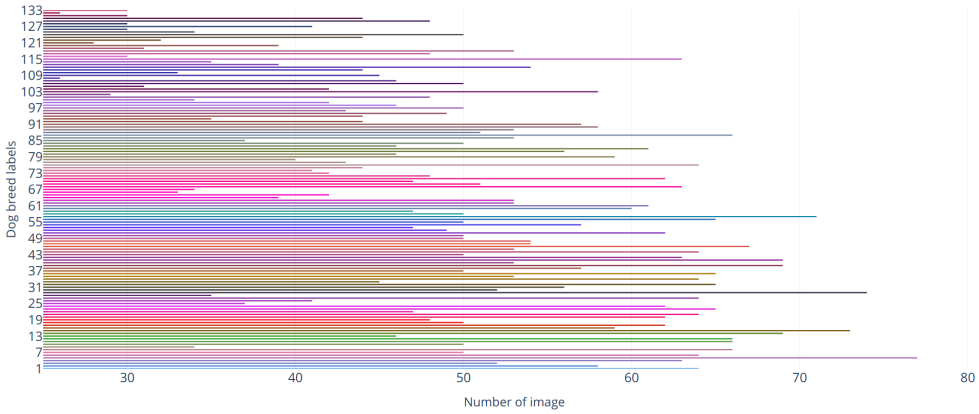

Figure 1: Distribution of training dataset

We also included two sample images from the dataset showing the images with dog only and with humans.

Figure 2: Dog only

Figure 3: Dog and human

4 Solution statement

We will create a CNN model for the dog breed classification. I will use several convolutional layers, each followed by a Max Pooling2D to reduce complexity of the stacked model. We will convert each feature map in the Max Pooling Layer into a scalar. The model will be based from scratch and will output the dog breed. Also, I will consider using Rsenet50 library for improving feature engineering and transfer learning for the dog breed classification.

5 Benchmark model

Pre-trained VGG-16 model can be a benchmark model. VGG-16 model, with weights that have been trained on ImageNet (a well-known very large and popular dataset used for image classification and other vision tasks). ImageNet

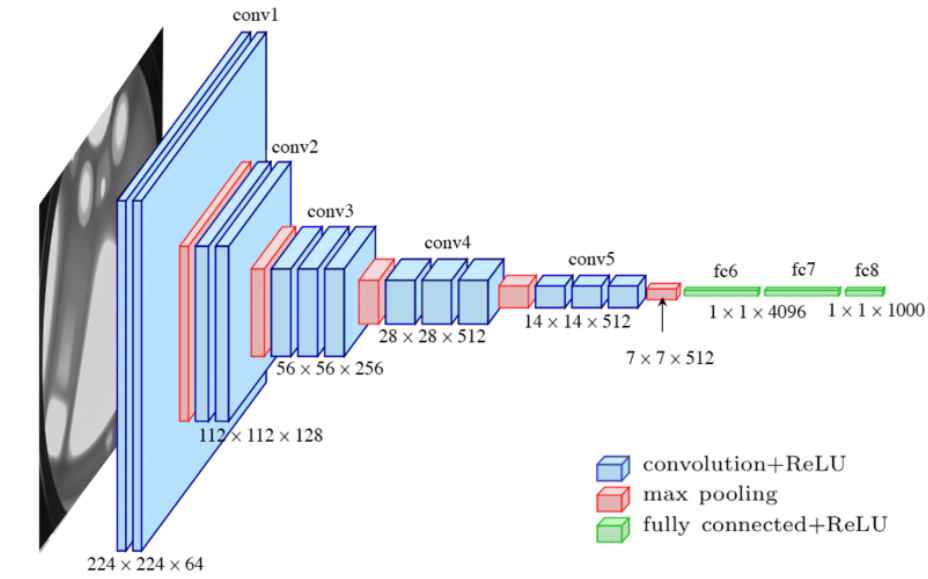

Figure 4: Architecture of VGG16

contains over 10 million URLs, each linking to an image containing an object from one of 1000 categories. For the pre-trained VGG-16 model, we need only check if the pre-trained model predicts an index between 151 and 268 (inclusive), indicating a dog detection.

6 Evaluation metrics

Evaluation metrics include the accuracy, and the rounds of the iterations (training speed). Since the data is unbalanced. We will also consider F1 score as a metric.

7 Project design

For this project, we will first explore the data and try out the classifiers for the dogs and humans. We will use OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces and apply VGG16 to dog detection. WE will apply customized-built CNN model to train the dog images for dog breed classification. Finally, We will apply Rsenet50 and transfer learning to compare the performance of the custmized CNN model and the transfer learning results. All the images will be normalized. And the training, test, and validation dataset provided will be directly used for training, testing and validation of the

model. Also, we will use rotation, horizontal flip and center crop etc. techniques to help augment the data.

Reference

Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).