Machine Learning Engineer Nanodegree

Capstone Proposal

Minh Gia Hoang

October 18, 2020

Dog Breed Classifier using CNN

I. Domain Background

Dog breed classifier is a representative image classification problem in the computer vision (CV). Classifying dogs into their breeds are often challenging even for human by simply looking at them. Thus, it needs to be done by computer instead. Solving this problem is useful when rescuing dogs, finding them homes, treating them, etc. or just simply seeing a cute dog and wondering which breed the dog belonged. Given an input image, the first task is to identify if it is a dog image or a human image. If it is a dog image then classify it to one breed of dog. If the image is a human face then identify the most resembling dog breed associated with the face.

Convolutional neural networks (CNN) have been widely used for their efficiency to deal with 2-D data such as images. Thus, CNN will be employed to accomplish this project.

Besides, some different approaches have been explored for this dog breed classifier. The table below summarizes the benchmarks of some early attempts on the Stanford Dogs dataset (Fei-Fei, 2011).

Table 1: Summary of Benchmarks of Stanford Dogs (Hsu, 2015).

Method	Top – 1 accuracy
SIFT + Gaussian kernel (Fei-Fei, 2011)	22%
Unsupervised learning template (Shapiro, 2012)	38%
Gnostic fields (Han, 2015)	47%
Selective pooling vectors (Kanan, 2014)	52%

II. Problem Statement

The objective of the project is to build a pipeline that can be used within a web or mobile app to process real-world, user-supplied images. There are two main tasks:

- Human face detector: If given a human face image, model will predict the most resembling dog breed associated to face.
- Dog breed detector: If given a dog image, model should predict its breed.

III. Datasets and Inputs

The input type for the project must be image type. Udacity provided datasets to train, validate, and test models. In particular, they comprise of images of dog breeds and humans in separate directories. One contains 13233 images of human faces while the other includes 8351 images of dogs. Variations in image sizes throughout the datasets are high.

The dog dataset is structured into a training set, a validation set, and a test set. Each set comprises 133 breeds of dogs organized into subdirectories by breed name. Note that the number of images per dog breed is imbalanced. There are 6680 dog images in the training set, 835 images in the validation set, and 836 images in the test set.

IV. Solution Statement

To solve the multi-class classification problem in CV, CNNs have been widely adopted. CNN can capture the spatial dependencies of pixels in an image by applying 2-D convolution. The architecture is suitable for image datasets since a small set of parameters (the kernel) is used to compute outputs of the entire image, so the model has much fewer parameters compared to a fully connected layer. Put differently, CNN can be trained to understand the sophistication of the image better than feedforward neural networks. The solution involves 3 stages:

- Human face detection using OpenCV's implementation of Haar feature-based cascade classifiers.
- Dog detection using the VGG-16 model, along with weights that have been trained on ImageNet.
- Dog breed classification using first a CNN built from scratch, then trying residual neural network (ResNet) e.g., ResNet50 is a good starting point or pre-trained model from ImageNet competition to significantly models boost the accuracy to meet the requirements of the project.

V. Benchmark models

A CNN is built from scratch and should get more than 10% accuracy. This accuracy can confirm that the model is working since a random guess yields a correct answer about 1 in 133 times, which corresponds to an accuracy of less than 1%.

The pre-trained model should achieve an accuracy of more than 60% so it can be used in a dog breed classifier app.

VI. Evaluation Metrics

Due to class imbalance in the dataset, besides accuracy, precision and recall would be appropriate choices for evaluating the proposed model.

VII. Project Design

The workflow for the project to approach the solution is as follows:

1. Import necessary python packages and libraries. Import and preprocess the data and split it into train, validation and test sets.
2. Detect human faces using OpenCV's Haar cascade classifiers.

3. Detect dogs using pretrained VGG16 model.
4. Create a CNN to classify dog breeds from scratch. Train, validate and test the model. Test accuracy should be greater than 10%.
5. Improve the accuracy by transfer learning. Test accuracy should be greater than 60%.
6. Combine all the model.
    - 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.

VIII. Bibliography

Fei-Fei, A. K. (2011). Novel Dataset for Fine-Grained Image Categorization. *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition.* Colorado Springs, CO.

Han, G. C. (2015). Selective Pooling Vector for Fine-Grained Recognition. *IEEE Winter Conference on Applications of Computer Vision*, (pp. 860-867).

Hsu, D. (2015). *Using Convolutional Neural Networks to Classify Dog Breeds.* Stanford University.

Kanan, C. (2014). Fine-grained object recognition with Gnostic Fields. *IEEE Winter Conference on Applications of Computer Vision*, (pp. 23-30).

Shapiro, S. Y. (2012). Unsupervised Template Learning for Fine-Grained Object Recognition. *Advances in Neural Information Processing Systems 25*, (pp. 3122-3130).