Dog Breed Classifier Proposal

Domain Background

Convolutional neuron network(CNN) has drawn increasing interest among machine learning field[1]. CNN is a class of deep neuron network, which provides a promising platform in the area of image and video recognition, image and video recognition, recommender systems, image classification, medical image analysis, natural language processing, and financial time series[2]. However, traditional deep learning neuron network which is implemented in fully connected multilayer perceptron, usually require millions of parameters to be trained and prone to overfitting. Even with fast growing computation capability to date, it's still very challenged to train those fully connected deep neuron network. Whereas, CNN exploit a sparse local connectivity pattern between neurons of adjacent layers: each neuron is connected to only a small region of the input volume. Scale of connectedness and complexity will drop tremendously and make training possible. In this work, a dog breed classifier will be implemented based on CNN.

Problem Statement

In this work, a pipeline will be built to process real-world, user-supplied images. Given an image of a dog, the algorithm will identify an estimate of the canine's breed. If supplied an image of a human, the code will identify the resembling dog breed.

Datasets and Inputs

Udacity dog and human datasets will be used in this work.[3] The human dataset consists of 13233 human images and with human name included in image file name. All images are in size of 250X250 and stored in 5750 folders sorted by human name alphabetically. Each person can have more than one images, which means the data is not balanced. Besides, the dog dataset has 8351 total images which have been split into train, test and valid directories. There are 133 dog breeds to be classified which correspond to 133 folders in train, test and valid directories.

Solution Statement

In this work, firstly, a human face detector will be implemented. To be specific, a pre-trained face detectors based on Haar feature-based cascade classifier[4] will be deployed directly. Secondly, the first dog detector has been implemented via transfer learning. A pre-trained VGG-16[5] will be used to make predictions of whether the image is a dog or not. After that, a CNN will be created from scratch to classify dog breeds. Then, use pre-trained model such as VGG-16 to classify the dog breeds and compare with the self-implemented CNN classifier from previous step. Finally, an app algorithm will be implemented to test the above algorithms.

Benchmark model

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman in 2014. This model achieves 92.7% in ImageNet[6], which is a dataset of over 14 million images belonging to 1000 classes. Among those classes, 137 dog breeds from 'Chihuahua' to 'Mexican Hairless' are included. This makes VGG16 as a good candidate for transfer learning with last 2 fully connected layers retrained.

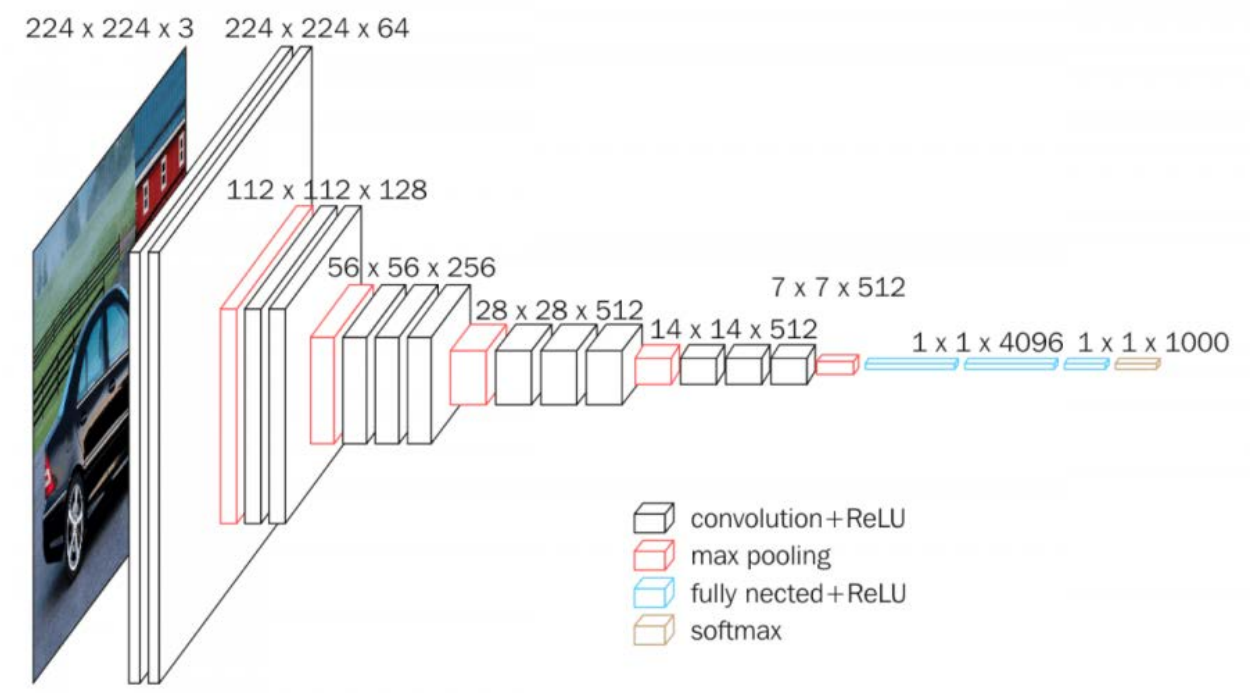

Figure 1. VGG16 architecture [7]

Evaluation metrics

Accuracy will be the main metric in this work, which can be calculated as:
Accuracy = (TP + TN) / (TP + TN+ FP + FN),
where TP is true positive, TN is true negative, FP is false positive and FN is false negative, respectively.
The scratch CNN model should generate an accuracy beyond 10%, although it's not very high, but much better than random guessing which is ~ 1%. High performance CNN classifier such as VGG16, requiring weeks of training. Due to the limit of computation resources and time cost, the metric target for scratch CNN model from this course project is reasonable.
Transfer learning model based on VGG16 should surpass 60%.

Project Design

Step1: Download the datasets and pre-process the data set into train, test and validation data sets. All image should be reshaped and normalized before training.

Step2: Import the human detector model implanted based on Haar feature cascade classifier.

Step3: Import pre-trained VGG166 as a dog detector

Step4: Create a CNN from scratch to for dog breed classifier. Train and validate the model at the same time and save the model when validation loss starting to surpass training loss; in this way, we will reduce overfitting issues. This CNN generation accuracy should be larger than 10%.

Step5: Taking advantage of pre-trained VGG16, freeze all convolutional layers and leave the last 2 fully connected multiple linear classifier layers unfreeze and retrained based on provided dog dataset.

Step6: Write an App algorithm for test. Given an image of a dog, the algorithm will identify an estimate of the canine's breed. If supplied an image of a human, the code will identify the resembling dog breed.

Reference

1. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.
2. https://en.wikipedia.org/wiki/Convolutional_neural_network
3. https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/dogImages.zip
4. https://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html
5. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
6. http://www.image-net.org/
7. https://neurohive.io/en/popular-networks/vgg16/