Dog Breed Classifier Report

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. Since many great CNN architectures are proposed to solve computer vision problems. Dog breed classifier will be an ideal candidate to study 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. It's a typical machine learning classifier question.

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. Here we can see each input image is a daily life picture and can even include background which will make the problem even more challenged.

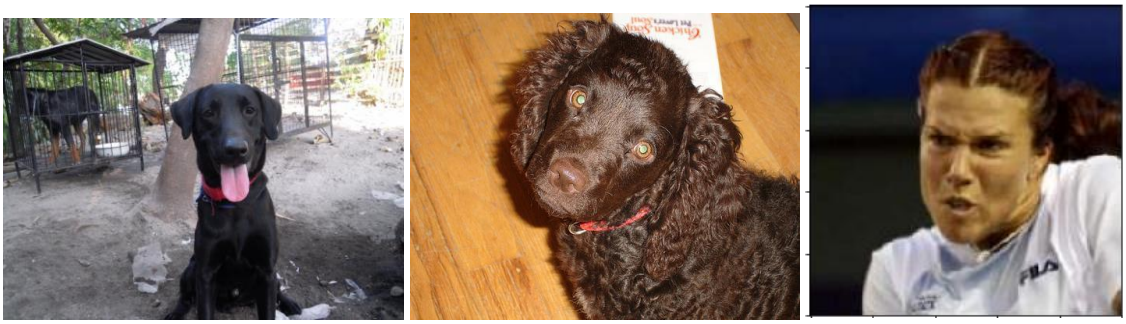


Figure 1. Image samples for dog and human in datasets

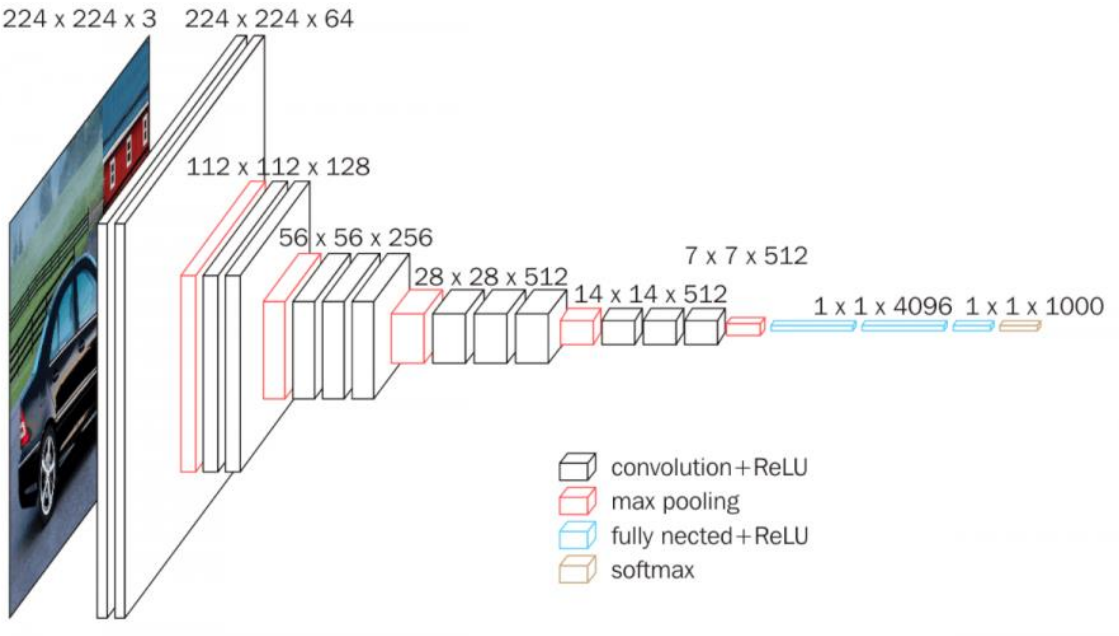


Figure 2. VGG16 architecture [7]

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%.

Algorithms and Techniques

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 as showed in Fig 2, 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.

Data Processing

Images are resized to 224X224 for all data sets. For training set, image augmentation has been done to improve performance of the classifier. Image augmentation techniques used consists of random crop, flip and rotation. Then images will be normalized among all data sets for better training efficiency.

Implementation

As mentioned above, the human detector was directly imported from OpenCV. Dog detector was a direct deployment of VGG16. Our in-house model will be based on CNN. It consist of 3 convolutional layers each has a 3X3 filter size with stride 1. After convolution with the kernel filter, 2X2 max pooling layer will be executed to reduce the dimensions by 2 on each side. For activation, ReLU will be used. Then a 3X224X224 image input tensor will be converted into 64X28X28 after 3 convolutional layers. Then 2D array will be flatten into 1D vector and mapped to 500 nodes by a FCC. 25% random nodes will be dropped out to avoid overfitting by getting rid of some high weighted neurons. Then a batch normalization will be implemented after activation for better performance. The second FCC will map the vector from hidden layer to final classification which ranges from 1 to 133 corresponding to a dog breed from Affenpinscher to Yorkshire_terrier.

Transfer learning has been implemented based on a pre-trained VGG16. All parameters are frozen expect the last two fcc layers since our classes number are not the same as VGG16's original dataset problem. The last two fcc layer will be re-trained for our dog dataset.

Refinement

For the scratch CNN model, techniques such as batch normalization has been introduced. It brings the accuracy from 7% to 13%, then fit our target range.

Model Evaluation and Validation

Human face detector demonstrated good regeneration rate. 98% of human faces were detected in the first 100 images of human data sets. On the other hand, dog detector provides 100% accuracy for the first 100 images of dog data sets. Scratch CNN model after hyper parameter tuning, demonstrates 13% accuracy. Furthermore, transfer learning based on VGG16 provides 84% accuracy. Both training and validation has been done in the same time. In general, training loss will decay along with running epochs. Due to overfitting, validation loss will first decrease then increase with training epochs. The model will be saved at the epoch when validation loss reaches the minimum.

Justification

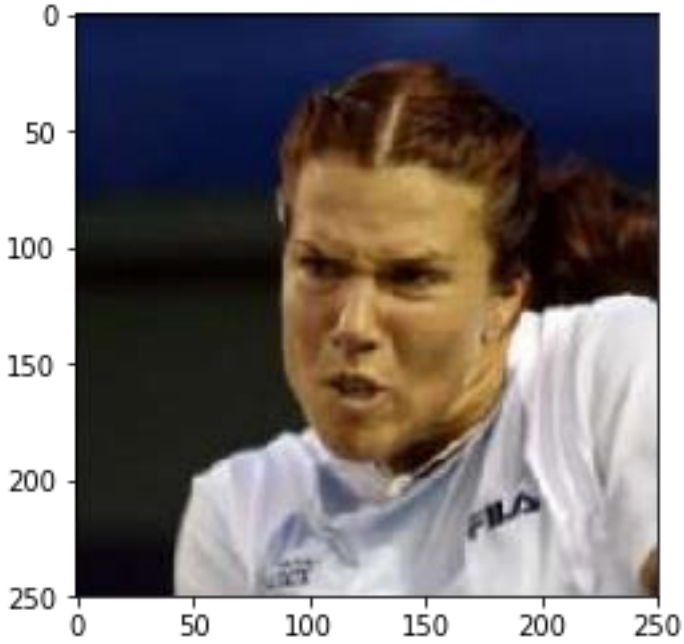
All models implemented in this work has satisfied our goals. The scratch CNN model surpass the 10% target with accuracy 13%. Transfer learning model provides 85% accuracy which is also larger than proposed 60% model.

Finally, demo of the dog breed classifier has shown in below.

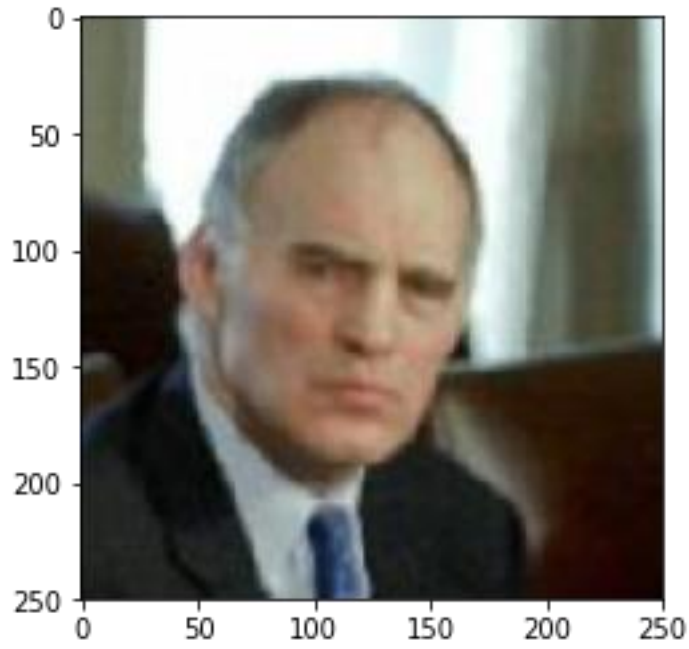

```
dog detected and breed is  Lakeland terrier
```


```
dog detected and breed is  Canaan dog
```

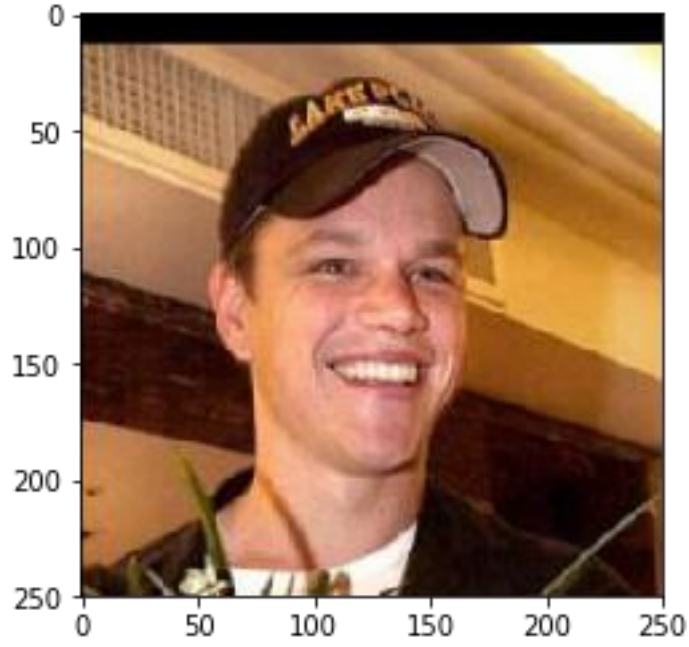
dog detected and breed is Greyhound



Hello Human!
 You look like a ... Australian shepherd




```
Hello Human!
 You look like a ... Basenji
```




```
Hello Human!
 You look like a ... English springer spaniel
```

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/