Tim Lin Machine Learning Capstone Project
August 19, 2020

Dog App Report

Project Overview

Image recognition has been around since the 1960's aiming to mimic human vision for the computer to tell object in the image. And only recently, with pre-trained Conventional Neural Networks (CNNs) and ImageNet, reaches a breakthrough in performing this task to the public. The pre-trained models help users save time to train a model from scratch and service as a benchmark and foundation for the developing model. ImageNet has helped provide over 14 million tagged images over 1000 classes as a source of data to develop and design a more sophisticated computer vision algorithm.

Conventional Neural Network (CNN) in deep learning, a branch of Machine Learning, is inspired by the biological process in the connectivity pattern between brain neurons resembles the organization of the animal visual cortex. Individual neurons only respond to specific receptive fields. The receptive fields partially overlap with each other created an entire visual field.[1]

In practice, a CNN consists of an input layer, an output layer, and multiple hidden layers. The hidden layers mostly contain a series of convolutional layers, pooling layers, fully connected layers, and normalization layers. The convolutional layer convolves the input and passes its result to the next layer. The pooling layer reduces the dimensions of the data but retains the information contained in them for less expense computation. The normalization layer re-center and rescale the input layer to speed up and reduce the sensitivity to network initializations. Usually, at the end of the network, there is a fully connected layer that connects every neuron in all layers to one another layer.

Several well-known CNN architectures are LeNet, AlexNet, GoogLeNet, VGGNet, and ResNet. LeNet is the first successful application of Convolutional Networks developed by Yann LeCun in the 1990s used to read zip codes, digits. And for this project leverages pre-trained VGGNet and ResNet on ImageNet for image recognition. Karen Simpnyan and Andrew Zisserman developed VGGNet demonstrated that the depth of the network is critical for model performance. Their best network contains 16 layers, also known as VGG16, with a stack of convolutional layers, followed by three fully-connected layers achieved 92.7% top-5 test accuracy in ImageNet.[2] Kaiming He et al. developed Residual Network which substantial use of batch

normalization and unique skip connections without fully connected layer at the end of the network.[3]

Problem Statement

The purpose of this project is to create the backend of a web application to classify between human and dog breeds and suggest the dog's breed or human resembled dog breed.

Metrics

- Accuracy for binomial classification of humans and dogs with 100 samples from each category.
- Log Loss is used for dog breed classification, considering there are 133 classes and takes into account the uncertainty of the prediction based on it varies from the actual label.

Dataset Exploration

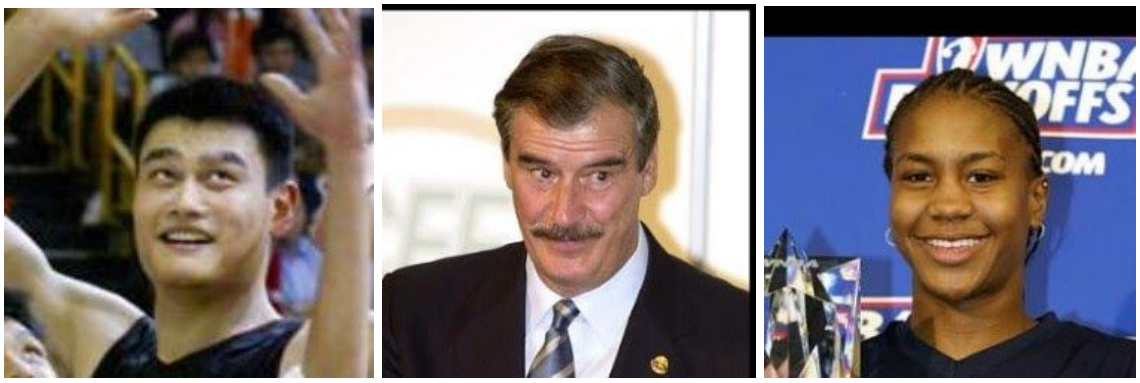
The input of this project is images provided by Udacity stored in https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/dogImages.zip and https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/lfw.zip.

The *dogImages* folder contains a total of 8,251 images in three subfolders, test (836 images), train (6,680 images), and valid (835 images). Each of the subfolders contains 133 directories with each dog's breed in different directories. Sampled images various in sizes and have different backgrounds. And the number of a dog's image is imbalance across different breeds.

The *lfw* folder contains 13,233 human face images in 5,750 subfolders, one folder for each person's name. Sampled images are all the same size and centered with different backgrounds. And the number of images for a person is not the same.

Sample Pictures for Dogs

Sample Pictures for Human



Algorithms and Techniques

A Convolution Neural Net (CNN) is used for multi-classes image classification. The model takes an input image on one end and output class scores at the end. The first step is to classify images for humans or dogs. OpenCV's implementation of the Haar feature-based cascade classifier is used to detect the human face, and the pre-trained VGG16 model is used to detect dogs. Then, depending on the model performance, either a CNN model from scratch or a pre-trained ResNet is used to predict the dog breed if it is actually classified as a dog bread or a dog breed that a human resembled based on the model.

Benchmark

1. A CNN model from scratch to classify dog breeds with at least 10% accuracy.
2. A transfer learning CNN (ResNet16) to classify dog breeds with at least 60% accuracy.

Data Preprocessing

All dogs' images are resized to 256x256 and cropped to 224x224 at the center to prevent loss information. For training images, augmentations (random horizontal flip and random rotation) are used to reduce overfitting. Then, all images are converted to tensor with normalized means and standard deviations of ImageNet.

Implementation

A CNN model is built from scratch to predict a dog's breed. The model consists of an initial convolution follow by three layers of Convolutions, Max Pooling, and Rectified Linear Unit Activation Functions. Then, two Fully Connected Layer (FCL) with in-features as for the first layer and its output as the input for second FCL with the output as the number of classes. Two 25% dropouts are used before each FCL to

prevent over-fitting. The model takes in a pre-processed 224*224 image and output a dog's breed category. The pre-trained ResNet might be used if the accuracy is low. Cross-Entropy loss function and Stochastic Gradient Descent optimizer, with a learning rate of 0.01, are used to train all models in 20 epochs.

Refinement

The CNN model from scratch has an accuracy of 10%, which barely meets the benchmark. Therefore, the plan changed to use an alternative model with a pre-trained ResNet18 and a Fully Connected Layer outputs 133 scores, one for each dog breed. The model achieved 80% accuracy, which bypasses the benchmark, and it is far better performance than the model from scratch.

Model Evaluation and Validation

- OpenCV's Human Face Detector
    - True Positive (human detected as a human): 98%
    - False Positive (dog images detected as a human): 17%
    - Overall accuracy: (98+83)/200 = 90.5%
- VGG16's Dog Detector
    - True Positive (dog detected as a dog): 100%
    - False Positive (human detected as a dog): 0%
    - Overall accuracy: (100+100)/200 = 100%
- Transfer Learning ResNet16 Dog Breed Classification
    - Overall accuracy: (669/836) = 80%
    - Loss: 0.67 at 20 epoch

Justification

Both the human face detector and dog detector models achieved high overall accuracy, so the models are suitable. The transfer learning model achieved an 80% overall accuracy, bypasses the benchmark by 20% at the first iteration. The model is safe enough for the app, but there could be improvements.

Improvements

- Train the existing transfer learning model for more epochs because when training the model, there is always a validation loss decrease observed from each of the 20 epochs that ran.
- Use a pre-trained model with more layers, such as ResNet50 or ResNet152.
- Increase the size of the training dataset since some breeds only have a few images.

Reference

1. Matusugu, Masakazu; Katsuhiko Mori; Yusuke Mitari; Yuji Kaneda (2003). "Subject independent facial expression recognition with robust face detection using a convolutional neural network" (PDF). *Neural Networks*. 16 (5): 555–559. doi:10.1016/S0893-6080(03)00115-1.
2. https://neurohive.io/en/popular-networks/vgg16/
3. https://cs231n.github.io/convolutional-networks/#norm