Dog Breed Classifier

Machine Learning Nanodegree Project Proposal

Jiawei Sun

Domain background

According to the study of Internet Center(IDC), 70% of information nowadays is transferred via image or video[1]. With the development of computer technology in the field of image classification, we are now able to formulate object detection approaches, train a computer to classify and recognize images and enable computers to better serve humans[2]. Traditional machine learning models are designed for a limited number of samples, which leads to insufficient performance and generalization ability[3]. Artificial neural networks have shown an outstanding performance in handling classification and recognition tasks. Convolution neural network(CNN), one of deep learning frameworks, is proved to have advantages in speed and accuracy. In this project, I will design a "Dog Breed Classifier" based on convolution neural network framework.

Problem statement

The goal of the project is to develop a deep learning algorithm that can be used to distinguish dog breed for a given image. For a user-supplied image, the classifier should first identify whether the face in the image is a dog or not. If it's an image of a dog, the algorithm should return with an estimated dog breed. If a human is detected in the image, the algorithm should identify a dog breed that the human resembles.

Datasets and inputs

The classifier in this project accepts images as inputs. The dataset is provided by Udacity and contains 13,233 images of humans and 8,351 images of dogs. The human dataset will be used when we use OpenCV's implementation of Haar feature-based cascade classifiers to find human faces. The dog dataset will be used to when we use pre-trained VGG-16 model to detect dogs in images. At the end of the project, we will test out this algorithm using six images that I uploaded.

Solution statement

In this project, Convolution neural network(CNN) is applied to perform a multiclass classification. Developing the classifier is a four-step process. First, we use OpenCV's

implementation of Haar feature-based cascade classifiers to detect human faces. Second, we use pre-trained VGG-16 model to detect dogs in user-supplied images. Then, a CNN model will be developed from scratch to classify dog breeds. This model should perform a test accuracy higher than 10%, which is set by Udacity. Finally, a transfer learning will be used when we apply ResNet 50 model to greatly improve the accuracy of the CNN model. It is required to achieve a test accuracy higher than 60% set by Udacity.

Benchmark model

The CNN model created from scratch should attain a test accuracy of at least 10%. The CNN pre-trained model with transfer learning should attain significantly enhanced accuracy. This model is required to reach accuracy higher than 60% to be used as a dog breed classifier app.

Evaluation metrics

Accuracy will be the key metric in this project to evaluate the performance of the benchmark model and the final solution. Accuracy is defined as the percentage of correctly classified instances. In the equation as shown below, TP, FN, FP and TN represent the number of true positives, false negatives, false positives and true negatives, respectively.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Project design

Step 1: Import dataset and libraries.

Step 2: Detect human faces by applying OpenCV's implementation of Haar feature-based cascade classifiers. We use pre-trained face detector to extract the number of faces founded. If the number is greater than zero, then we assume there is a human face on the image.

Step 3: Detect dog faces by using the pre-trained VGG-16 model. It will check if the face belongs to one of all dog classes, then reach a conclusion whether the image has a dog or not.

Step 4: Create a CNN model from scratch, classify images and achieve accuracy of at least 10%. Apply resizing, cropping and augmentation on the images. Train, validate and test the model.

Step 5: Create a CNN by using pre-trained weights from a ResNet-50 model. Test accuracy should be higher than 60%. First, pre-process data by applying resizing, cropping and augmentation on the images to prevent from overfitting and increase data variety. Second,

use the weights learned by convolutional layers in ResNet-50 model, and then train only the fully connected layers to develop the dog breed classifier. Finally, use Cross Entropy Loss, so that the model will be trained based on an increasing number of epochs until the loss stops dropping.

Step 6: Test out the model with images from the Udacity datasets and the images uploaded by me.

References:

[1].Jun-e Liu, Feng-Ping An, "Image Classification Algorithm Based on Deep Learning-Kernel Function", Scientific Programming, vol. 2020, Article ID 7607612, 14 pages, 2020. https://doi.org/10.1155/2020/7607612

[2].Neha Sharma, Vibhor Jain, Anju Mishra, An Analysis Of Convolutional Neural Networks For Image Classification, Procedia Computer Science, Volume 132, 2018, Pages 377-384, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2018.05.198.

[3].Xin, M., Wang, Y. Research on image classification model based on deep convolution neural network. J Image Video Proc. 2019, 40 (2019). https://doi.org/10.1186/s13640-019-0417-8