Domain Background

Image classification has been an object of study since 1950s and it was a tremendous challenge because of the complexity of the task. It is vital for many current and future applications such as quality control, radiography, self-driving cars and even agriculture.

To compare different methods scientists participate in the competitions like the ImageNet Large Scale Visual Recognition Challenge or Pascal VOC before that. In 2012, a convolutional neural network (CNN) won the ImageNet Challenge for the first time and that was a breakthrough. In the following years, CNNs won every year and proved to be an excellent tool for image classification.

This breakthrough in deep neural networks became possible thanks to constant progress in CPU and especially GPU performance that we see in the last decades. Assuming this growth in computing power is highly likely to continue, CNNs will become more and more affordable and widespread, which underlines the importance of studying CNN's applications.

Problem Statement

The goal is to build a machine-learning model that tells us the breed of a dog on a given photo. In addition, if it is a photo of a human it tells us which dog breed it resembles the most.

We have a dog breed dataset so we are doing a reinforcement learning and this dataset has 133 breeds so it is a multi-class classification problem.

This problem consists of few steps that allows us to understand different aspects of image classification and try different methods.

1. Detect humans
2. Detect dogs
3. Classify dog breeds using CNN created from scratch
4. Classify dog breeds using CNN created using transfer learning

Datasets and Inputs

For this project, Udacity provided images of dogs and humans.

There are 8351 dog images divided in train, test and valid folders. We have 133 directories (one for each dog breed) in each of train, test and valid folders. The images are of different sizes, different backgrounds, some images are full-sized, and some are of just a head. The data is not balanced because the number of images for each breed varies.

All human photos are sorted by person's name. There are 13233 images in 5749 folders. Human images are all of the same size 250 by 250. Images are with different backgrounds and light, from different angles but just heads no full-size pictures. The data is not balanced because the number of images for each person varies.

Solution Statement

Convolutional Neural Networks (CNN) are the best algorithms today to classify images according to the results of the ImageNet Challenge. CNN have a significantly different architecture than other artificial neural networks. CNNs use a concept of hierarchy where first convolutional layers detects the most basic elements of an image and pass that knowledge along to the next layer. Then each next layer detects more and more complicated features of the picture.

This hierarchical nature of CNN allows us to build a model in two possible ways: from scratch or using pre-trained model. The latter is called transfer learning and it is particularly interesting because training a big CNN from scratch could be very computationally intensive. State-of-the-art CNNs, competing in the ImageNet Challenge are usually very large and require super-computers to train. So, the ability to use such pre-trained model and train only few last layers of the network to solve our specific problem is very promising.

Benchmark Model

The CNN model created from scratch must have accuracy of at least 10%.
The CNN model created using transfer learning must have accuracy of 60% and more.

Evaluation Metrics

In this project, we will use accuracy to evaluate our model.

Accuracy is the ratio of number of correct predictions to the total number of input samples.

We will also use training loss and validation loss to evaluate the training process and change model's parameters when needed.

Project Design

Step 1. Import the necessary datasets and libraries, pre-process the data and create train, test and validation datasets. Perform Image augmentation on training data.
Step 2. Detect human faces using OpenCV's implementation of Haar feature based cascade classifiers.
Step 3. Create dog detector using pre-trained VGG16 model.
Step 4. Create a CNN to classify dog breeds from scratch. Train and test the model.
Step 5. Create a CNN to Classify dog breeds using Transfer Learning. Train and test the model.
Step 6. Write an algorithm to combine dog detector and human detector:

- If human is detected in the image, return the resembling dog breed;
- If dog is detected in the image, return the predicted breed;
- If neither is detected, provide output that indicates the error.

References

https://github.com/udacity/dog-project

https://en.wikipedia.org/wiki/Convolutional_neural_network

https://pytorch.org/docs/stable/index.html

https://www.kaggle.com/c/dog-breed-identification/overview/description

http://rajatvikramsingh.github.io/media/DeepLearning_ImageNetWinners.pdf