DOG BREED CLASSIFIER

Machine Learning Nanodegree Capstone Proposal

Satya Prakash Biswal

Domain Background

In this project I am going to develop an algorithm to classify dog breeds. I will create a pipeline that will process user-supplied images and try to classify them into one of the 133 breeds using deep learning algorithms [3].

Image classification is one of the most important fields in computer visions. One of other most important challenge is the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) where each year new classification algorithms are reviewed [3,4]

Problem Statement

To write an algorithm that accepts a file path to an image and first determines whether the image contains a human, dog, or neither.

- Detect dog and predict breed
- Detect human face, return resembling dog breed
- Detect neither

Datasets and Inputs

The datasets to be used: dogs dataset and human dataset.
The dogs dataset is composed by:

- Total images: 8351
- Classes: 133

human dataset

- human images: 13233

All the images are resized to 244x244 and normalized before being used with the model.

Solution Statement

Since this is multiclass image classification problem, CNNs are appropriate candidates for the algorithm.
The solution of the project follows six main steps.
First, we need to explore the datasets in order to understand how to use them and choose the proper algorithms. Second, implement a Haar feature-based cascade classifier using OpenCV in order to detect faces in the human dataset.
In the third step, I will use a pre-trained VGG16 model in order to detect dogs in the dogs dataset
Fourth, I will create a LeNet[5] like architecture that uses CNN in order to classify the 133 dogs breeds and have an accuracy greater than 10%. In step five, I will use the transfer learning technique in order to a pre-trained ResNet50 architecture and continue the training with the dogs dataset. The minimum accuracy required is 60% on the test set. Seventh, I will write a webapp that accepts a file path to
an image and first determines whether the image contains a human, dog, or neither.
In the last step, I will Test the Algorithm with some random images found online.

Benchmark Model

The solution model was compared to a benchmark CNN with:
- 4 convolutional layers, each of them followed by ReLU and 2D Max Pooling,
- a flattening layer,
- a drop-out layer,
- a fully-connected layer,
- ReLU,
- drop-out
- and another fully-connected layer.

Evaluation Metrics

In order to deal with a multi class classification problem, the evaluation metric that will be used is the negative log-likelihood loss function.
Using this metric, the performance of both models will be evaluated by using the accuracy metric: number of correct predictions divided by total number of predictions. We can also compare recall, precision and F1 scores to compare different models.

Project Design

The solution designed for this project will follow these steps:

- Step 0: Import and explore dog and human data sets
- Step 1: Detect Humans using a Haar feature-based cascade classifiers
- Step 2: Detect Dogs using a pretrained network trained on ImageNet VGG-16 model
- Step 3: Create a CNN to Classify Dog Breeds (from Scratch) using a LeNet like architecture.
- Step 4: Create a CNN to Classify Dog Breeds using Transfer Learning and a using a ResNet50 architecture.
- Step 5: Write a custom Algorithm that accepts a file path to an image and first determines whether the image contains a human, dog, or neither.
- Step 6: Test the Algorithm with some random images

References

1. http://www.fci.be/en/Nomenclature/Default.aspx
2. https://en.wikipedia.org/wiki/Dog_breed
3. Olga Russakovsky*, Jia Deng*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. (* = equal contribution) **ImageNet Large Scale Visual Recognition Challenge**. *International Journal of Computer Vision*, 2015.
4. Marcel Simon, Erik Rodner, Joachim Denzler**, ImageNet pre-trained models with batch normalization**, *ArXiv*, 201
5. https://en.wikipedia.org/wiki/LeNet