Project Proposal

Dog Breed Classifier
Author: Minh Tuan Dang

Domain Background

According to the database of The Fédération Cynologique Internationale is the World Canine Organisation, 368 dog breeds exist [1]. Modern dog breeds formation was driven by dog shows in the late 19th century [2]. Breeds classification remains a relevant problem for the dog owners who searches for a show-class puppy or for a working dog with particular performance characteristics. Deep Learning has been proved to be suitable for image classification problems [3], therefore the task can be tackled by a Deep Learning algorithm like CNN or RNN technology

Problem Statement

The primarily investigated problem is dog breed classification via Convolutional Neural Network (CNN). The project also covers the tasks:

- Dog face detection
- Human face detection

Datasets and Inputs

1. Dog dataset [4]. The `dogImages/` folder contains 133 folders, each corresponding to a different dog breed.
   There are 8351 total dog images.
2. Human dataset [5] for testing on human images.
   There are 13233 total human images.

The datasets were included into the project workspace on Udacity. Alternatively, the datasets could be downloaded via the provided links and unzipped locally.

The human dataset was used for testing of the dog breed classifier in such case that it predicts which breed a human look like. The dog dataset was used for training, validation and testing of the CNN classifier.

The datasets are appropriate given the context of the problem.

Solution Statement

The solution addresses the multiclass image classification problem. Since the classified objects are images, CNNs are appropriate candidates for the algorithm. CNNs are a type of Deep Learning algorithms that can take tensors (e.g. batches of images) as an input, establish the relations between image features by learning weights and biases, output a predicted class and learn from an error.

The proposed solution, first, classifies the input images as dog, or human, or none of the two categories. Second, if a dog or a human is present on the image, the app predicts one of 133 breeds.

The base of the predictor is VGG-16 with batch normalization trained on ImageNet [6]. The final classification layer was replaced by another dense layer with 133 neurons instead of 1000 (the number of ImageNet classes [3]).

The model was trained for 10 epochs, optimized by Stochastic Gradient Descent with the learning rate = 0.001 to reduce the Cross-entropy loss function. The performance during training was validated based on the loss value on the validation set. On the test set, the model achieved 84% of accuracy and the loss of 0.538.

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.

The architecture was designed to follow the common structure of CNN classifiers and VGG in particular: the first layers are convolutional layers, the number of features increases in higher layers. The feature extractor is followed by flattening of the feature tensor and the classifier: 2 dense layers with activation functions. The kernel size of (3, 3) is the most popular, having a number of features as a power of 2 is also a standard.

The benchmark model was trained for 51 epochs, optimized by Stochastic Gradient Descent with the learning rate = 0.005 to reduce the Cross-entropy loss function. The performance during training was validated based on the loss value on the validation set. On the test set, the model achieved 28% of accuracy and the loss of 2.90087.

Evaluation Metrics

The performance of both models was evaluated by using the accuracy metric: number of correct predictions divided by total number of predictions. This metric was mentioned by the exercises' designers. Accuracy is one the widely used metrics for multiclass classification evaluation.

Project Design

The project was conducted in 7 steps:

- Step 0: Import Dog and Human datasets
- Step 1: Detect humans by OpenCV's implementation of pre-trained Haar feature-based cascade classifiers.
- Step 2: Detect dogs by trained on ImageNet VGG-16 model.
- Step 3: Create a CNN from scratch to classify dog breeds. The CNN is described in details in the section *Benchmark Model.* Train and test it.
- Step 4: Create a CNN to classify dog breeds using Transfer Learning. Train and test the solution model (consult *Solution Statement* for details)
- Step 5: Design an algorithm that accepts a file path to an image and first determines whether the image contains a human, dog, or neither. Then,
  if a dog is detected in the image, return the predicted breed.
  if a human is detected in the image, return the resembling dog breed.
  if neither is detected in the image, provide output that indicates an error.
- Step 6: Test the algorithm on sample 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. https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/dogImages.zip
5. http://vis-www.cs.umass.edu/lfw/lfw.tgz
6. Marcel Simon, Erik Rodner, Joachim Denzler**, ImageNet pre-trained models with batch normalization**, *ArXiv*, 2016.