7 May 2020
Marianna Janczak

Capstone project

Machine Learning Engineer Nanodegree Project Proposal

CNN project: Dog Breed Classifier

1. Domain background

Mobile apps making use of machine learning algorithms are becoming more and more popular. They often utilize speech recognition and computer vision to offer richer and more user friendly experience. They can help people with disabilities as well as provide entertainment.

As I have always been interested in the fun part of using machine learning, I decided to take on a project, that can join education with entertainment and prepare a model, that is able to recognize dog breeds from pictures. Additionally, it should be able to detect humans on the provided pictures and suggest the most similar dog breed.

This is a Udacity specific project, which means that is has an already prepared data set and a set of instructions, guiding through different possible implementations and exploring the subject of convolutional neural networks. The main repository for the project can be found on GitHub[1].

2. Problem statement

This project aims at constructing and training a machine learning mechanism, that is able to classify images. Any appropriately preprocessed image, given to the mechanism should result in one of three possible outputs:

a) if the image is a picture of a dog, the mechanism should output information about the breed of the dog

b) if the image is a picture of a human, the mechanism should suggest what dog breed is most resembling the shown human

c) in any other case the mechanism should inform user, that the image contains neither a picture of a dog nor of a human.

3. Datasets and inputs

As this is a Udacity specific project, the dataset for it has already been provided. There is no indication, as to the source of the images. The dataset consists of two parts:

a) dog dataset - 133 folders containing images, each corresponding to a different dog breed

b) human dataset - over 13 thousand images of humans

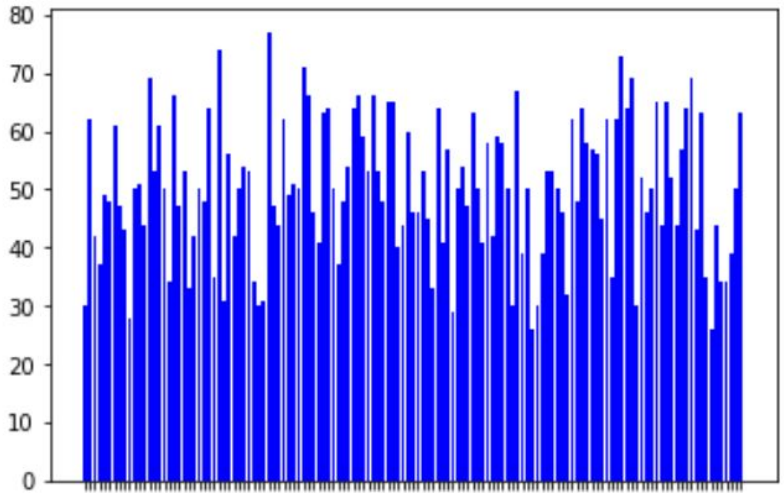
Both human and dog images will have to go through a suitable preprocessing. After that, the human images will be used to train a human face detector with the ue of OpenCV[2] library and the dog images will be the input for classification models, based on CNNs and transfer learning.

The dog dataset is divided in three parts - the main one of 6680 images for training, 835 for validation and 836 reserved for testing purposes.

The training data is rather balanced among the 133 classes, with the average of 50 images per class, the maximum being 77 and minimum 26. Below is the histogram showing the distribution of the training data among the 133 classes:

[1] https://github.com/udacity/deep-learning-v2-pytorch/tree/master/project-dog-classification

[2] https://docs.opencv.org/trunk/index.html



4. Solution statement

There are actually distinct parts of the problem, that will require an appropriate solution:

a) dog detection and dog breed classification - based on CNN and transfer learning (transfer learning is needed, as the dataset is relatively small and would not be enough to train a well performing model from scratch).

b) human face detection - here an OpenCV library will be used

c) finding a dog breed most resembling given human face - the dog breed classifier will be used

d) creating a mechanism that will put all the previously mentioned parts together and give the required output - a simple python function will do the necessary calls to the previously trained models.

5. Benchmark model

As a benchmark a pretrained VGG-16[3] model, that is available as a part of PyTorch library[4] will be used. The model is pretrained on ImageNet[5]. This is a large image dataset, separated into 1000 classes, indexed 0-999. Images with different dog breeds are in classes with indexes 151 and 268 (inclusive), which allows to use it as a basis for a dog detecting model.

For comparison purposes also a simple CNN model will be built and trained from scratch. This is to confirm, that witch such a small dataset transfer learning will be required.

6. Evaluation metrics

To quantify the performance of the models, accuracy will be used. As the project in reality consists of two separate models (one for human detection and one for dog breed classifier), both models will be evaluated separately. Accuracy is the most basic metric that can be easily applied to classification problems. It takes into account both true positives and true negatives.

$$accuracy = \frac{(true\ positives + true\ negatives)}{dataset\ size}$$

7. Project design

The project implementation will be separated into several distinct steps:

a) Data preparation - the images will need to be preprocessed, to be suitable for the models. A preprocessing pipeline will need to be established, so that after training the model, the prediction can be performed on any chosen image.

b) Constructing a human detector with the use of OpenCV library. The human images will be used to train the model.

[3] https://arxiv.org/abs/1409.1556

[4] https://pytorch.org/docs/master/torchvision/models.html

[5] model. http://www.image-net.org/

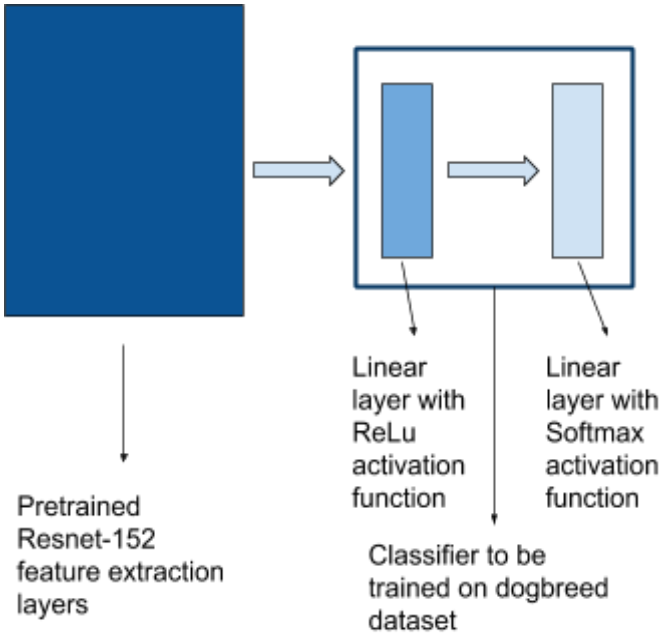
c) Setting up and running the benchmark model - VGG-16 pretrained on ImageNet

d) Setting up a reference model, creating a CNN model from scratch and training it on available dataset.

e) Creation of the main dog breed classifier model. This will be done with the use of transfer learning. *"Transfer learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned."*[6] This is motivated by the fact, that with a relatively small dataset training a model from scratch to recognize 133 distinct classes would not be effective. There have been attempts at solving this kind of task with CNNs, detecting facial keypoints and feature extraction, as presented in "*Dog Breed Identification*" paper by Whitney LaRow, Brian Mittl and Vijay Singh[7]. But their best classifier, as they write, "*predicts the correct dog breed on its first guess 52% of the time; 90% of the time the correct dog breed is in the top 10 predictions*".

As the base model a pretrained Resnet-152[8] will be used. Resnet models were designed to cope with the problems involved with really deep neural networks, like the vanishing gradient problems. Therefore they are well suited to cope with many convolutional layers needed to extract features from images. The pretrained Resnet-152 made use of the ImageNet dataset, which ensures that the convolutional layers were properly trained to extract features from images. To make use of this powerful tool, only the last layers (the classifier) of the original model will be exchanged for the layers trained with the dog breed dataset.

Proposed architecture:



8. References

1. Udacity Dog Breed Project Github Repo
https://github.com/udacity/deep-learning-v2-pytorch/tree/master/project-dog-classification
2. OpenCV https://docs.opencv.org/trunk/index.html
3. VGG-16 https://arxiv.org/abs/1409.1556
4. PyTorch pretrained models https://pytorch.org/docs/master/torchvision/models.html
5. ImageNet dataset model. http://www.image-net.org/
6. Chapter 11: Transfer Learning, Handbook of Research on Machine Learning Applications, 2009

7. "Dog Breed Identification" paper by Whitney LaRow, Brian Mittl and Vijay Singh
https://web.stanford.edu/class/cs231a/prev_projects_2016/output%20(1).pdf
8. Resnet-152 model https://arxiv.org/abs/1512.03385

[6] Chapter 11: Transfer Learning, Handbook of Research on Machine Learning Applications, 2009
[7] https://web.stanford.edu/class/cs231a/prev_projects_2016/output%20(1).pdf
[8] https://arxiv.org/abs/1512.03385