Capstone Proposal – Machine Learning Engineer Nanodegree
Laura Hartzheim, 24.11.2020

Dog Breed Classifier (CNN)

Domain Background

Image Classification is a common supervised learning task. Convolutional neural networks are usually applied to this problem. A very popular use case is to identify, wether an image shows a cat or a dog. In this project photos will be categorized by dog breeds. This is also a common task, as it has previously been part of a kaggle competition where 8715 entries were submitted by 1282 teams. (kaggle: https://www.kaggle.com/c/dog-breed-identification)

Problem Statement

The goal of this project is to create a web app, that allows users to upload their pictures for classification. If the image contains a human or a dog, a dog breed will be predicted, if not an error message should be displayed.

Dataset and inputs

The input used to create the algorithm will be in the form of images of dogs and humans. The dataset is supplied by udacity. It is split into two folders. The 8351 dog images are already split into training, test and validation subsets. It contains 133 classes. The human dataset includes 13233 photos. It does not have to be split into training, validation and test set, because it will be used on a pretrained classifier. In the final algorithm both datasets will be combined to test the complete workflow. The human dataset contains about 5000 pictures more than the dog dataset, which will lead to an imbalanced combined dataset. In the final web app users can upload their own photos as input.

Solution Statement

The first step will be to to use a pretrained classifier to detect humans in pictures. Next another pretrained net will be used to check wether an image contains a dog. To identify the specific breed of a dog a CNN will be used. First a net will be created from scratch. This CNN should at least have an accuracy of 10%. Later the results will be improved using transfer learning. This should lead to an accuracy of at least 60%. The final step will be to combine all previous results. If a dog or human is detected, a dog breed will be predicted, if not an error message will be displayed.

Benchmark Model

The benchmark model for this project will be the CNN for dog breed classification that will be created from scratch. It is supposed to have an accuracy of at least 10%. This should be high enough to ensure the model is not just making random guesses.

Evalutation Metrics

To measure the quality of the model the accuracy of its predictions will be calculated. Some benchmarks are already mentioned in “Solution Statement”. The formula used to calculate the accuracy is: accuracy = 2*(precision * recall) / (precision + recall). These metrics will work with the imbalanced dataset.

Project Design

The following steps will be taken during the execution of this project:

1. Import Datasets

   The two datasets have to be downloaded for further steps.

2. Detect Humans

   A pretrained model from OpenCV that uses haar cascades will be used to detect if a picture contains a human face. The model returns a bounding box around the face. This output will be processed to “true” if a face could be found in the picture and “false” if not.

3. Detect Dogs

   A VGG-16 model that was trained on the ImageNet data will be used to detect wether an image contains a dog. The VGG-16 can detect 1000 classes. In this project only the dog related classes are needed to detect if the image contains a dog. Here the output will be processed into “true” and “false” again.

4. Create a CNN from scratch to detect dog breeds

   Here a dog breed classifier will be implemented from scratch using pytorch. It should have an accuracy of at least 10%. The model architecture will be inspired by VGG-16 (https://neurohive.io/en/popular-networks/vgg16/), however the aim is to create a smaller net. Another inspiration for the architecture will be ResNet (http://www.ai-united.de/eine-uebersicht-ueber-resnet-und-seine-varianten/).

5. Use transfer learning to train a CNN to detect dog breeds

The CNN created in this step should have an accuracy of at least 60% on the test set. The pretrained model used for transfer learning will either be VGG-16 or ResNet.

6. Combine models from previous steps to one algorithm

   In this step the results from the previous steps will be combined. First the algorithm will detect if the image contains a dog or human. If not the output should be an error message. Otherwise one of the dog breed detection nets will be used to detect a dog breed for the image. The output will be displayed on the web app.

7. Test the algorithm

   The complete algorithm will be tested with at least six images that are not part of the datasets that were used during the implementation of this project. If the results are not satisfying some rework needs to be done.