Capstone proposal

Dog breed classification project

Franziska Zimmermann

Domain background:

Identification and classification of objects in images is one of the most classical application areas of machine learning. Over the last decades neural architectures used to solve this task have taken an impressive evolution. Over time new publications brought in more and more smart ideas to process image information yielding better performances on benchmarks datasets like ImageNet. Some important milestones in this process of evolving cnn network architecutres are LeNet (LeCun et al., 1998), AlexNet(Krizhevsky et al., 2009), VGGNet (Simnoyan et al., 2014), GoogLeNet
(Szegedy et al., 2014), ResNet (He et al., 2015), DenseNet (Huang et al., 2016) and ResNeXt(Saining et al., 2017). Since I'm passionate about computer vision applications of machine learning algorithms I chose the Dog Breed classification project to explore how humans and dogs can be detected in images and how transfer learning can be used to predict the breed of a dog in an image reliably although this is a task that most humans (that are not dog experts) would fail in. At first glance the problem of classifying dog breeds might seem not too relevant to solve in reality, however the classification of objects that vary only slightly in appearance (for example MRI images of different types of tumours) is in general a very important problem. Due to the availability of sufficient training data in this project the classification of dog breeds was chosen however knowledge gained during project development will surely be useful for more relevant problems in the future. Moreover this project allows me to understand the challenges involved in piecing together a series of models designed to perform various tasks in a data processing pipeline and using its individual strengths and weaknesses in a smart combination.

Problem statement:

The goal of this project is to build a pipeline to process real-world, user-supplied images. Given an image of a dog, the algorithm will identify an estimate of the canine’s breed. If supplied an image of a human, the code will identify the resembling dog breed. So as a first step, the algorithm has to decide whether a human or dog is present in a given image. In a second step it then decides about the breed of the apparent dog or the most resembling dog breed for a human.

Datasets and inputs:

The datasets of humans and dogs were provided by Udacity and unfortunately I couldn't find any details about their origin. It contains 13233 human and 8351 dog images. The human images folder contains portrait photos of several prominent figures (sometimes only one picture per person, sometimes several). The dog images folder is split into a train, dev and test subdirectory. Each of these contains 133 subfolders for the various dog breeds. Each dog breed subfolder itself contains many different photos showing dogs from the respective breed. The photos have varying backgrounds and show the dogs sometimes in full-body mode, sometimes just in portrait mode. There are even some pictures that also contain a human or multiple dogs. The dataset is sufficiently large and

contains enough variation for training a model that can generalize well to real-world images not seen during training.

Solution statement and project design:

For solving the task of detecting human faces in images OpenCV's implementation of Haar feature-based cascade classifiers will be used. For identifying whether a dog is present in an image a Pre-trained VGG 16 net (trained on ImageNet) will be used. The performance of both models will be evaluated on the given datasets.
For classifying dog breeds, I will first try to train a small CNN from scratch. For that I will investigate necessary/useful pre-processing steps for training and test images and selection of appropriate optimizer, loss function and training hyper parameters. The trained model will be evaluated on the given test set. Of course with this approach (training from scratch) we can't expect a very good accuracy of the trained model since the chosen network architecture will be probably too simple to represent this complex task. Of course we could invest in more time and resources for training a larger model from scratch but this is not necessary: Luckily there are several pre-trained bigger models available that contain lots of useful information in earlier layers that can be reused for the given problem. Therefore as a next step, transfer learning will be applied to get a better performing model for classifying dog breeds. Several pre-trained architectures will be finetuned on the given dataset of 133 dog breeds and evaluated on the test set. For performing transfer learning in a lightweight manner I will use Pytorch Lightning and will finetune a pre-trained Mobilenet, Resnet50 and ResNext101. Of course this again includes appropriate preprocessing of the images, selection of appropriate loss functions and optimizers and training hyper parameters.
Finally I'll put together the algorithms for identifying human faces, dogs and classifying dog breeds to build the final application and test it on some images the algorithm has never seen during training.

Benchmark model:

As a benchmark model, I will train a really small and simple CNN architecture (consisting of a view conv and max pooling layers) from scratch. I will evaluate it on the test set and show that it yields a classification accuracy much better than a random guess, however a lot too small to be an acceptable solution for the given problem.

Evaluation metric:

The problem whether a human or dog is visible in a given image is a binary classification problem which can be evaluated by looking at the accuracy of the binary classifier. The problem which dog breed is shown in a picture of a dog is a multi-class (and single-label) classification problem which can be evaluated by measuring the accuracy of the classifier as well. Since distribution of samples among the 133 classes is approximately equal and classification results on all classes are equally important plain accuracy does the job and we don't have to think too much about macro f1 or precision/recall on special classes.