Machine Learning Engineer Nanodegree

Capstone Proposal

Gabriel Verzea
September 3rd, 2018

Proposal

Domain Background

One of my favorite topics of the ML Engineer nanodegree is image recognition and identification. Early in the program, I have thought of using the techniques I have learned, to develop an app that would help me, and others like me, with 2 other topics of interest: long distance running and foraging. Many times on my long trail runs I have run short on food and struggled to finish the distance, low on energy. Also, I have always appreciated my mother for knowing so well all the wild plants and fruit, but I never managed to learn them myself – they all look the same to me!

So when I ran into an internet article depicting some wild edible plants, I know I found my project – image recognition – later making also a mobile app for it, that would recognize which of the plants I encounter on my runs are good to eat, and also offer information on which parts of the plant are edible, and which are not, or maybe even dangerous if ingested.

A similar computer vision problem is described in the article [Wäldchen, J. & Mäder, P. Arch Computat Methods Eng (2018) 25: 507](). Here, a number of plants are recognized mainly based on leafs and flowers, using a number of features as Shape, Color, Texture and Leaf.

Problem Statement

The classification problem to be solved is to identify if a given picture is that of one of the 62 edible wild plants included in the dataset. The algorithm should return either one of the 62 plants, with the related additional information, with the highest resemblance to the given picture, or decide if the given picture depicts none of the plants in the dataset.

Datasets and Inputs

I have searched for a dataset that would help with my problem, but I could find none. I did find a lot of pictures on only 3 of my plants in an [existing Kaggle dataset](#), and I have obtained permission to include them in my dataset. For the rest of the plats, I have myself collected pictures off the internet. I have gathered a starting number of 50 pictures for each plant, and plan to add more as I advance with the project, especially for the classes where the algorithm will not yield good results.

The images are in .jpg format. The dimensions are not manually standardized, but this will be one step of the data preparation – to bring all the images to 100x100px. So far, I have made sure that no image exceeds 300KB, to keep the database within reasonable size limits. The color profile of a JPG image is YCbCr, which is a mathematical coordinate transformation from an associated RGB color space, according to [Wikipedia](#).

The gathered pictures will be used to train/test the model. To preserve the class balances across the train/test datasets, I plan to use the [Stratified K-Folds cross-validator](#). At the end, the model will be tested both with pictures set aside for testing, as well as new pictures that would contain, or not, one of the 62 edible wild plants that the model has been trained on. The dataset I have put together is uploaded [here](#).

Solution Statement

The solution I propose is using deep learning, to be more specific, CNNs, to help a model learn a number of features about the training pictures, so that it would yield a precision of at least 60% on the test dataset.

I would use Keras, use transfer learning and start from the VGG-16 model, use some pre-trained layers and weights, and only add an output layer of 62 nodes and train those weights.

Benchmark Model

To benchmark my model, I would use a vanilla (simple) CNN model.

Evaluation Metrics

The evaluation of both models will be done on Classification Precision (or misclassification error).

Precision = True_Positives / (True_Positives + False_Positives)

Generalizing the above to the multi-class with macro-averaging (average the performances of each individual class), the formula becomes:

Precision(macro) = (Precision(1) + … + Precision(k))/k

Project Design

In the beginning, I would need to create from the available dataset a list of the images, and read it. I would also read the additional information about the plants from an excel file.

As the dataset is gathered off the internet, the pictures come in different sizes and quality levels. At first I would need to pre-process the data. I would try the following actions:

- Obtain an uniform aspect ratio – by cropping to squares (centered)
- Perform Image scaling – scale to 100x100px
- Normalize image inputs – this makes convergence faster while training the network. Data normalization is done by subtracting the mean from each pixel, and then dividing the result by the standard deviation
- Perform Data Augmentation – generate more pictures from the existing ones, by rotation, scaling or flipping

After the data is preprocessed, I will use the pre-trained weights of a VGG-16 model and build the output layer (softmax of 62 nodes), and experiment with few additional intermediate hidden layers. This is ok as I want to predict a single label for an image.

The existing dataset will be split into training, dev and test datasets.

The model will be benchmarked against the selected benchmark model above, to assess its performance. This will be measured by a single evaluation metric - the macro-averaged classification precision.