Machine Learning Engineer Nanodegree
Capstone Proposal

Bas Donker
Date: 2018/01/10

Domain Background:
In the age of social media and smartphones that have better cameras than professional cameras less than two decades ago, we take a lot of pictures. We update our social media feeds with pictures of food, we send selfies over instant messages, and we even create separate social media accounts so we can justify uploading hundreds of photos of our dog.

Just two generations ago, taking a picture of something was an event in and of itself. Film didn't come cheap, so more care was taken into taking the perfect shot. Once the camera roll was filled up, it would be taken to a shop to develop the photos and these photos would then carefully be put into a photo album during a rainy Sunday afternoon.

In the far future, we can show these photo albums to the grandkids to show them what life was like before they were around.

With the amount of pictures we take today, an individual photo is significantly less important than a photo used to be. We take a few dozens of takes for a group photo, just in case someone blinked in any of them.

We certainly never take the time to organize our photos anymore. There's too many of them and we're too busy taking new ones to be able to spend time organizing. This also means it becomes very difficult to find a picture again after it's been buried under the thousands of new pictures we've taken since then.

The result of this is that, in general, all of our hundreds or even thousands of pictures all live together in one folder, completely unstructured and unorganized. Are you finding yourself in a conversation with your colleagues about hiking trips and you want to impress everyone with that great picture of you on the Preikestolen in Norway you took over 3 years ago? Guess you'll have to scroll through a seemingly endless list of food pictures and cat memes first. By the time you find it, the conversation will have moved to a different topic already.

This problem can be overcome by automatically structuring your photos based on the images themselves and what is in them. We can categorize them and either tag them so that they can be searched for, or moving them into different folder, so your camera roll is more structured.

According to this paper, a convolutional network can be implemented for this:
http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Problem Statement:
We will be taking a set of images and classifying them into different categories based on what objects are in the images. For instance, if the image contains rolling green hills, mountains, or grass fields and no people, we want to classify it as a landscape. If the image contains one face and it is my own, it should be in the selfies category. If it's a picture of multiple people close together all looking at the camera, it's a group picture.

Solution Statement:
We can use a convolutional neural network to look at a selection of pictures and determine what is in the picture.

As inputs, we will provide the convolutional neural network with images, along with their correct labels. The network will learn how to classify these images correctly and will then test its hypothesis on a different labelled set it hasn't seen before.

If the network is successful enough, we can run it on new data. It will label all the pictures in our camera roll, so that a different part of the application can move the pictures into the correct folders.

Datasets:
As the input data, I will be using pictures I have taken and received over the years. I will organize a few dozens of them manually, putting them in directories that correspond to the name of the label I want to give them.

These pictures have been taken with several different devices (different phones, an SLR camera, pictures I haven't made but I received from friends, etc.), so the resolution of these pictures vary significantly. Most, if not all pictures are in RGB colour.

Because of the differences in resolution and dimensions, the pictures will have to be pre-processed. This is because the convolutional neural network expects the input to be of a certain size, but also because training a machine on the original resolution will take far too long as it will go through each individual pixel.

We'll scale all the images down to a size of 250 by 250 pixels each, but we will maintain the RGB layers.

I will use categories the following categories:
- Landscapes
- City scapes
- Food
- Concerts
- Group photos
- My face

For each category, I will have around 100 images. These images will be augmented later and we'll be using transfer learning on a pre-trained model, so we can keep the dataset relatively small.

These images will be split up randomly in separate training, cross validation and test sets.

The images can be classified for multiple categories. If I'm in a group photo, I want the image to be classified as both my face and a group photo.

Benchmark Model:
To test the effectiveness of our Convolutional Neural Network, we will test its results against a traditional Fully Connected Neural Network and another algorithm that just makes random guesses as to which category a picture belongs to.

Evaluation Metrics:
If the model is working well, it's able to classify pictures it hasn't seen before and put them into its respective categories. Pictures of landscapes will be tagged with 'landscape', pictures of dogs will be tagged with 'dog', and selfies of my own face will be tagged with 'handsome bastard'.

The model will be evaluated based on their F1 scores.

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

The higher the F1 score, the better the model is performing.

Project Design:
In a directory on a computer, there are several subdirectories which are named according to the labels I want to classify my pictures as. For example, the file structure might look something like:

- Images
    - Landscapes
    - City_scapes
    - Food
    - Concerts
    - Group_photos
    - My_face

In the sub-directories will be pictures of landscapes, group photos, selfies and food respectively. During the pre-processing these pictures will be augmented. Copies will be made of them in which they are rotated a random amount, or they may be flipped horizontally. This is done to generate more data for the model to train on. We then resize all the pictures to 250 by 250 pixels.

After the data has been processed, we split them up randomly into training, cross validation and testing set.

We train the convolutional neural network to recognize the pictures and the categories to which they belong. We also train a conventional fully connected neural network to do the same, and we'll

write an additional model that just randomly guesses the category.

We can make different convolutional neural networks of different sizes and different amounts of convolutional layers.

For the transfer learning, we'll try different pre-trained networks:
- ResNet-50
- VGG-19
- Xception
- InceptionV3
- InceptionResNetV2

We then compare the results of all of these models to see which ones perform best on data they haven't seen before (the testing set).

Once we're happy with the model's performance we can implement it in the real world. I can create a new directory where I will upload all the pictures I take. The CNN will see that there are unclassified pictures in there and will attempt to put them in their rightful subdirectories.

This way I can more easily manage my pictures and I should be able to find specific ones without much effort at all.