Machine Learning Engineer Nanodegree

Capstone Proposal

Sergio Noviello September 11th, 2018

Proposal

Domain Background

Cervical cancer is a cause of death in women worldwide. In females is the 14th most common cancer. The mortality rates were drastically reduced with the introduction of smear tests, but in some countries is still very crucial to determine the appropriate method of treatment that is based on the position of the cervix. The wrong treatment will not work and will also be very expensive. This project is based on a competition that was launched on Kaggle a year ago and the reason I chose this project is that I strongly believe in the role that Machine Learning can play in the development of personalised cancer treatments.

Deep Learning and computer vision have been recently used in the healthcare domain for the classification of medical images.

In this paper, a group of researchers from Stanford University explain how neural networks can be built to classify cervical images. They used the same dataset I'm willing to use in this project. CS 231N Final Project Report (http://cs231n.stanford.edu/reports/2017/pdfs/924.pdf)

Another example is this publication on the National Center for Biotechnology Information Classification of Cervical Cancer Using Artificial Neural Networks (https://www.sciencedirect.com/science/article/pii/S187705091631170X)

Problem Statement

The problem I would like to solve is to develop an algorithm that is capable to classify cervical images and predict the probability of each image to be of type 1, 2 or 3.

This model will be very helpful for healthcare providers in order to identify patients with a cervix of type 2 or type 3 that require further testing.

Datasets and Inputs

The dataset is provided by Intel and MobileODT and it's available for download on Kaggle. (https://www.kaggle.com/c/intel-mobileodt-cervical-cancer-screening/data).

The train dataset consists of images organised in directories that represent their categories: Type_1, Type_2, and Type_3.

There are

- 251 images of Type_1
- 782 of Type_2
- and 451 of Type_3 for a total of 1484 images.

For the test dataseta all 513 images are in the same folder.

The images are all in jpg format and they are RGB images (channel red/green/blue). The dimensions are not always consistent. I will use opencv or Keras to resize the images.

Many images are showing a black area surrounding the cervix . During the pre-processing step I could try to detect those images and crop the area that contains valid information.

In order to reduce overfitting I will apply image augmentation techniques.

There is also an additional dataset provided with the archive additional*Type*{x}.7z. This contains images originally not included in the train set because of image quality or because they come from duplicate patients.

- Train set (https://www.kaggle.com/c/intel-mobileodt-cervical-cancer-screening/download/train.7z)
- Test set (stage1) (https://www.kaggle.com/c/intel-mobileodt-cervical-cancer-screening/download/test.7z)
- Test set (stage2) (https://www.kaggle.com/c/intel-mobileodt-cervical-cancer-screening/download/test.7z)

Solution Statement

I am planning to build a deep learning model using Keras. I will split the dataset into train and validation set. I will train the model on the train dataset and evaluate it on the validation set using the multi-class logarithmic loss.

I will finally use the model to classify the test dataset by assigning the probability for each class.

Despite the fact that the competition is closed it is still possible to submit predictions on Kaggle and be ranked on the leaderboard.

Benchmark Model

The benchmark model will be a simple convolution model. Then I will create a deeper network and increase the data using data augmentation.

I am planning to use transfer learning using pre-trained models like VGG16 or ResNet.

The winner of this competition on the private leaderboard achieved a score of 0.76963 (the closer to zero the better). The private leaderboard is calculated with approximately 75% of the test data.

Evaluation Metrics

For this Kaggle competition submissions are evaluated usig the multi-class logarithmic loss. For each image, a set of predicted probabilities (one for every category) needs to be calculated. The formula is :

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij}),$$

where N is the number of images in the test set, M is the number of categories, log is the natural logarithm, (y_{ij}) is 1 if observation (i) belongs to class (j) and 0 otherwise, and pij is the predicted probability that observation i belongs to class j.

Project Design

Data Preprocessing

In this part I need to collect the images from the train folder and test folder. As mentioned above some image contains black areas that could be removed by cropping the centre of the image that contains valid information.

I will also rescale the images. Some images are high pixel range, some are low pixel range. Scaling every images to the same range [0,1] will make images contributes more evenly to the total loss.

I will resize all the images to 200x200 using opencv.

I will augment the data using image augmentation. Image Augmentation is the process of taking images that are already in a training dataset and manipulating them to create many altered versions of the same image.

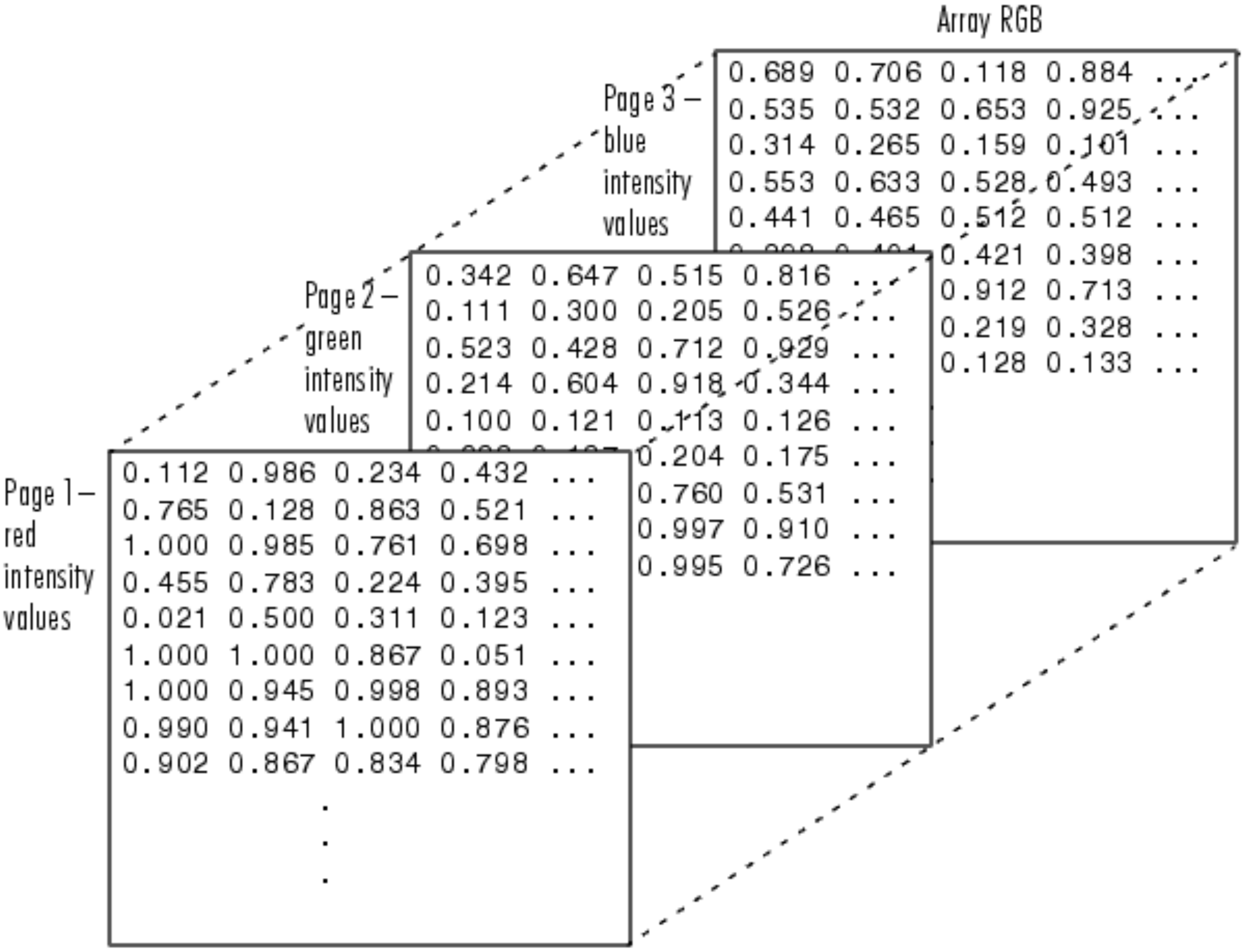
This is very simple to do in Keras using ImageDataGenerator

e.g.

```
ImageDataGenerator(rescale=1./255., rotation_range=25,
        height_shift_range=0.1, shear_range=0.2, zoom_range=0.2,
        horizontal_flip=True, fill_mode="nearest"
```

Representation of images as numpy arrays

An image can be represented as a multidimensional array. The dimensions are the width and height of the images and the number of channels, in this case 3 for the colors (red, green, blue). The values are the intensity of each color (red, green, blue) and are in the range from 0 to 255



Keras and opencv have methods that allow to convert images into arrays.

Validation split

Next step is to split the dataset into train and validation set. I will use 70% of the images for training and 30% for validation.

Build the Model

In this step I will use keras to build the neural network. I will use all the techniques studied in this course like regularization, batch normalization, dropout. If the results are poor I will consider transfer learning and use a pre-trained model like VGG16 or ResNet.

It's difficult to plan in advance how will be the architecture of the neural network. It's an iterative process that requires a lot of tweaking and adjustments.

I will use the ModelCheckpoint provided in Keras to check at the end of each epoch if the results are improving.

Evaluation

In this step I will have a model pre-trained and saved to a file. I will use the model to make predictions on the validation set and compare the type of cervix predicted by the model with the real label contained in the validation set. I will calculate the score using the multi-class logarithmic loss.

Prediction

In this step I will use the pre-trained model to make predictions on the test dataset and organize the predictions in a dataframe that will show the list of images and the probability of each class (type_1, type_2, type_3)

Example:

|image|type_1|type_2|type_3|

|0.jpg|0.003|0.54|0.67|

Score on Kaggle leaderboard

Last step is to submit the predictions to Kaggle to get a rank on the public leaderboard.

References

- kaggle competition (https://www.kaggle.com/c/intel-mobileodt-cervical-cancer-screening)
- cervix types classification (https://kaggle2.blob.core.windows.net/competitions/kaggle/6243/media/Cervix%20types%20clasifica
- keras documentation (https://keras.io/)
- log loss (https://github.com/ottogroup/kaggle/blob/master/benchmark.py)