Capstone Project

Cervical Cancer Screening

Definition

Project Overview

Since the inception of Machine Learning people have been applying it across various disciplines. One such domain where I think its application is making life changing difference is Medical Field. Some of the application of ML in Medical imaging are Diabetic retinopathy, Blood Flow Quantification, Tumour Detection etc. One of the problems in which I have liking is Cervical Cancer.

Cervical cancer is a cancer arising from the cervix. It is due to the abnormal growth of cells that can invade or spread to other parts of the body. Early on, typically no symptoms are seen. Later symptoms may include abnormal vaginal bleeding, pelvic pain, or pain during sexual intercourse.

Worldwide, cervical cancer is both the fourth-most common cause of cancer and the fourth-most common cause of death from cancer in women. In 2012, an estimated 528,000 cases of cervical cancer occurred, with 266,000 deaths. This is about 8% of the total cases and total deaths from cancer. About 70% of cervical cancers occur in developing countries. In low-income countries, it is one of the most common causes of cancer death. In developed countries, the widespread use of cervical screening programs has dramatically reduced rates of cervical cancer.

Source: *Cervical cancer - https://en.wikipedia.org*

Other references: *Denny L (2012) Cervical cancer: Prevention and T/t . Discover Med 14: 125-131, Ginsburg OM (2013) Breast and cervical cancer control in low and middle-income countries: Human rights meet sound health policy 1: e35-e41.*

Problem Statement

Cervical cancer is so easy to prevent if caught in its pre-cancerous stage. However, due in part to lacking expertise in the field, one of the greatest challenges of these cervical cancer screen and treat programs is determining the appropriate method of treatment which can vary depending on patients' physiological differences. Especially in rural parts of the world, many women at high risk for cervical cancer are receiving treatment that will not work for them due to the position of their cervix. This is a tragedy: health providers can identify high risk patients but may not have the skills to reliably discern which treatment which will prevent cancer in these women. Even worse, applying the wrong treatment has a high cost. A treatment which works effectively for one woman may obscure future cancerous growth in another woman, greatly increasing health risks. The solution to this problem is to provide the Health Care providers with a system which will determine Cervix type in real time, so that they can easily figure out patient's treatment eligibility based on it.

As a part of the solution I'll collecting the required data set, pre-process it and will train a classifier to predict the type given a cervix image.

Though the scope of this project is limited to creating a classifier only and not to create an end application.

Metrics

Use of ***Categorical Cross entropy Loss*** metric has been made to evaluate performance of the model.

Categorical Cross entropy Loss = $- \sum_{c=1}^{M} y_{o,c} * \log(p_{o,c})$

where ***M*** is number of classes (dog, cat, fish), ***log*** is the natural log, ***y*** is binary indicator (0 or 1) if class label ***c*** is the correct classification for observation ***o*** and ***p*** is predicted probability observation ***o*** is of class ***c***.

If the loss value is closer to zero, then the model is performing good. This metric is best suited when dealing with more then one class, it is also well suited when the distribution of data among the classes is uneven. One more reason for using this metric is that the benchmark model also uses this same metric for evaluating the performance.

Analysis

Data Exploration

The data set has been collection from Kaggle Competition *Cervical Cancer Screening - https://www.kaggle.com/c/intel-mobileodt-cervical-cancer-screening/data* Which is also the inspiration for this project. The data set consists of roughly 2000 images.

To determine the type of treatment fit for a Cervix cancer patient first we need to determine the type of cervix a patient has. Keeping that in mind data has been collected and has been divided into 3 types.

Figure 1: Type 1

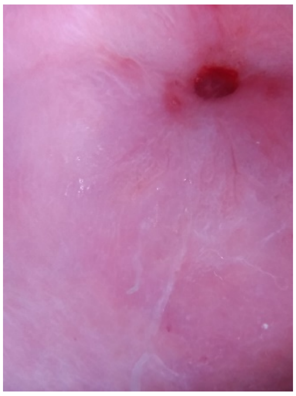


Figure 2: Type 2

Figure 3: Type 3

The dataset consists of collection of images of each type of cervix. These different types of cervix in data set are all considered normal (not cancerous), but since the *Transformation zones* aren't always visible, some of the patients require further testing while some don't. This decision is very important for the healthcare provider and critical for the patient. Identifying the *Transformation Zones* is not an easy task for the healthcare providers, therefore, an algorithm-aided decision will significantly improve the quality and efficiency of cervical cancer screening for these patients.

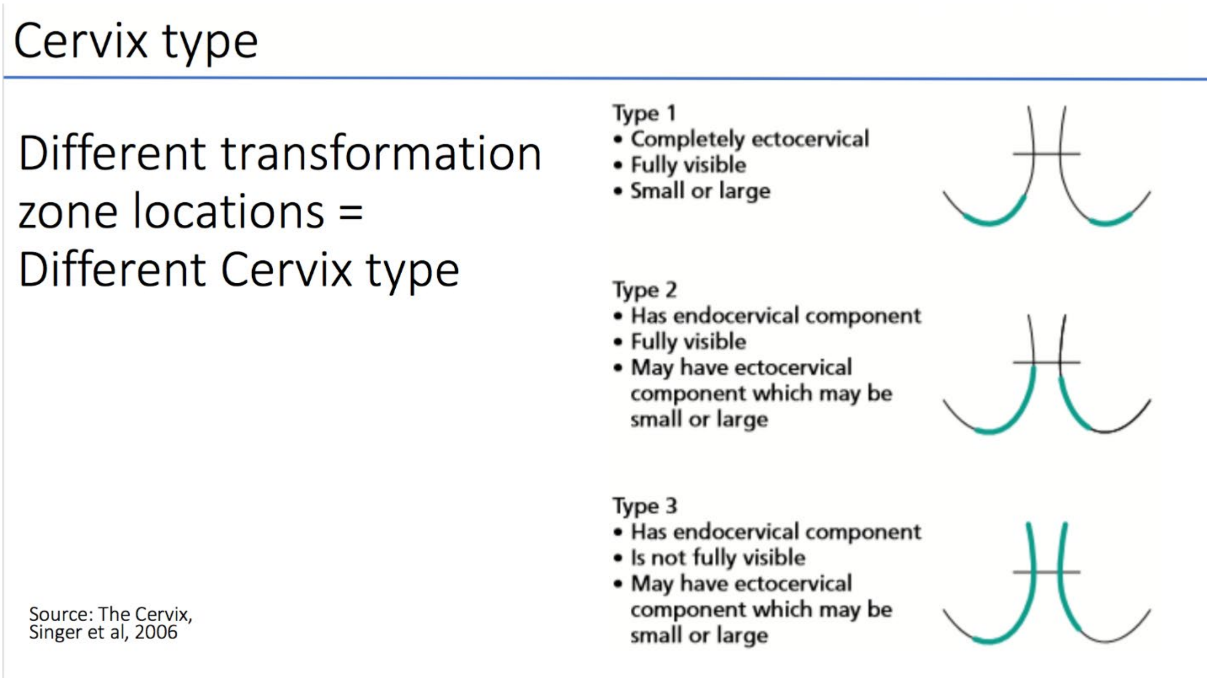
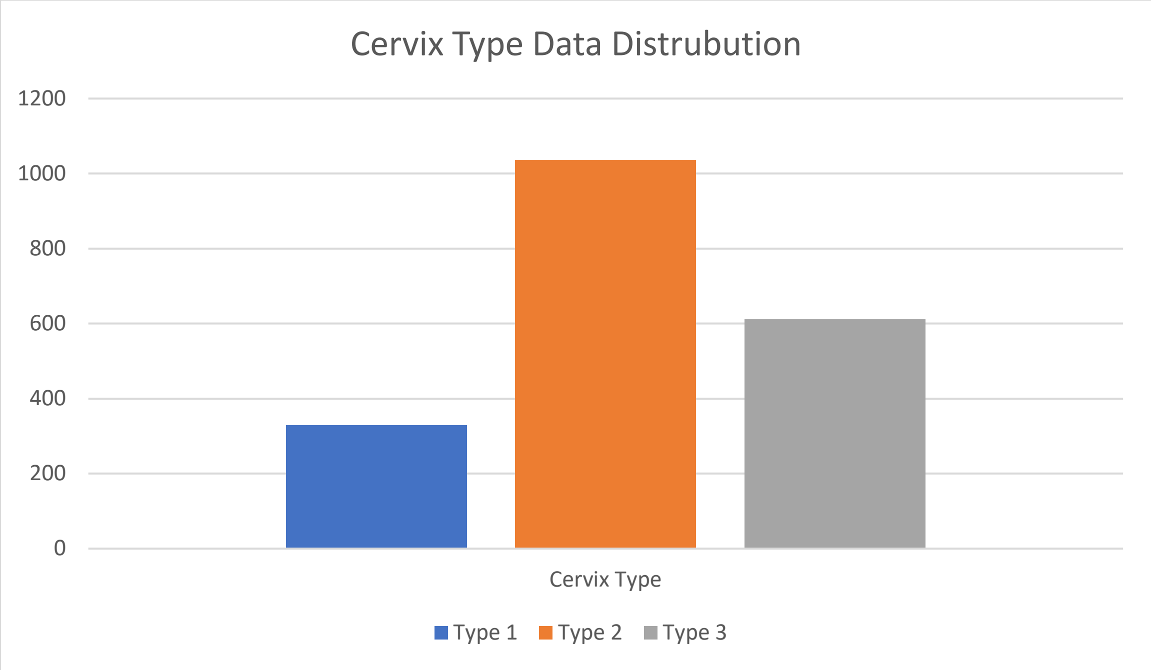


Figure 4

Exploratory Visualization



- Total samples in the dataset is 1975
- Images have dimension varying from 2000x3000 pixel to 3000x4000 pixel (approx.).
- The distribution of number of samples is very uneven as you can see from the visualization, here types 2 is leading with maximum number of samples.
- All the images are in JPG format.

Algorithms and Techniques

Convolutional neural net which is one of the best techniques of the current times for classification task has been used this project. They are popularly dubbed as CNN. CNN takes as input bunch of images in form of a tensor trains over them and finally given an image as input it give out class wise probability. CNN do really well with images, as it makes explicit assumptions that the input will be an image, and they have certain properties relating to image encoded in the architecture. The Network consist of 3 main types of layers, Convolutional which handles all the heavy computation stuff, Pooling which does the job of down sampling and Fully connected layer which gives us the output probability.

One of the important properties of CNN is that they are translation invariant, this means that even if the object of interest is little bit left or right or up or down in the other image, it will detect it without any special handling required. This property of the CNN is because of Max pooling layers. And my model will benefit from this, because while pre-processing I had to do some manual framing and while doing that, in some image's transformation zones got dislocated from the centre. So, I didn't have to handle them separately because of CNN's this property.

Other layer I have used is dropout which takes care of overfitting.

Hyper parameters that were tuned during the training:

- Batch size - Number of samples per gradient update
- Number of epochs – Training length
- Amount of dropout

As the data set size was small whole train and validation dataset was used throughout all the epochs.

As the data set is small, Transfer learning which a technique used when one has small data, is also used for the sake of comparison with my designed CNN.

A pre-trained VGG16 model has been used for transfer learning. Its fully connected layers were replaced with the new ones because the model was trained to classify 1000 classes and I my case I need to classify only 3. Further all the pervious Convolutional layer's weights has been frozen to take advantage of the pre-trained network. The network was trained on ImageNet dataset.

Benchmark

Current benchmark for the give problem statement is **0.70**, which is a multiclass loss value basically Categorical Cross entropy Loss, achieved by a team named "***Towards Empirically Stable Training***". They have made use of R-CNN models with VGG-16 feature extractors to achieve this feat.

Methodology

Data Pre-processing

A lot of pre-processing has been done out the Notebook as it required manual approach. The data set consisted of images which were either very blurry or the Cervix region was not clearly visible. Example the following:

Figure 6

Figure 7

This type of images was removed from the dataset. Further the rest of the image went through the manual framing processing, wherein I cropped all the images from test to train, to make cervix portion clearly visible.

Figure 8

Figure 9

Before pre-processing the average dimension was approx. 3000x4000 px. But after pre-processing the dimensions were following:

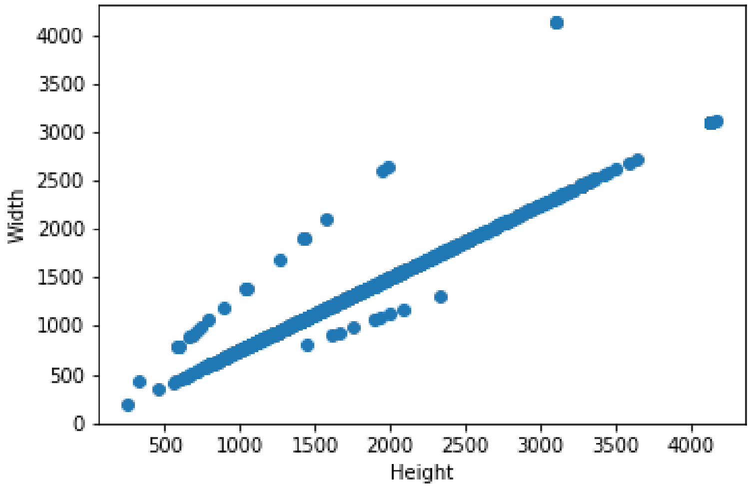


Figure 10

Implementation

The Architecture has been inspired from VGG16. Thought number of parameters to train in this network are much less than VGG16. Just like VGG16 it contains repeating sets of 2 convolutional and one max pooling layers. Here I have also introduced one dropout layer before max pooling one.

Strides in the first layer is kept to 2 and in remaining layers it is set to 1. *Padding* is kept to 'same' to keep the size of the output equal to that of the input to the layer. This setting allows us to go as deep as the hardware can support.

Activation function used in all the layers is 'relu' except for the final layer. Relu is the simplest of all the activation functions and provides more convergence than any other functions. In final layer Softmax function is used to get probability on multiple classes.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_9 (Conv2D)            (None, 256, 256, 32)      2432
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 256, 256, 32)      9248
_________________________________________________________________
dropout_1 (Dropout)          (None, 256, 256, 32)      0
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 128, 128, 32)      0
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 128, 128, 64)      18496
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 128, 128, 64)      36928
_________________________________________________________________
dropout_2 (Dropout)          (None, 128, 128, 64)      0
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 64, 64, 64)        0
_________________________________________________________________
conv2d_13 (Conv2D)           (None, 64, 64, 96)        55392
_________________________________________________________________
conv2d_14 (Conv2D)           (None, 64, 64, 96)        83040
_________________________________________________________________
dropout_3 (Dropout)          (None, 64, 64, 96)        0
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 32, 32, 96)        0
_________________________________________________________________
conv2d_15 (Conv2D)           (None, 32, 32, 128)       110720
_________________________________________________________________
conv2d_16 (Conv2D)           (None, 32, 32, 128)       147584
_________________________________________________________________
dropout_4 (Dropout)          (None, 32, 32, 128)       0
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 16, 16, 128)       0
_________________________________________________________________
global_average_pooling2d_2 ( (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 3)                 387
=================================================================
Total params: 464,227
Trainable params: 464,227
Non-trainable params: 0
_________________________________________________________________
```

Filter of very small size (3x3) is used. According to VGG16 paper(https://arxiv.org/pdf/1409.1556.pdf), this small filter size allows us to increase the depth of the network by adding more convolutional layer.

Finally, a global max pooling layer has been added to further reduce the number of the parameters to train and a dense layer to predict the class wise probability.

The pre-processed dataset was divided in to 3 parts Train 60%, Validate 20% and Test 20%. Train and Validate as name suggest are used for training and cross validation, while test dataset for calculating accuracy.

One major problem at hand was to decide upon the input size to the network. As the images were of varying size (Figure 10), I had to decide one size keeping in mind the amount of details from the image that will be require and the hardware that will be required had the images of such size. Keeping these 2 factors in mind and after few trails with different sizes (1500x1500), (1000x1000) and (512x512). I noticed that the images with (512x512) were giving just about the same accuracy as the other sizes.

Refinement

The initial model was proving loss close to 0.87 and an accuracy close to 46%. To improve dropout layers were introduced with dropout rate of 0.2. and the number of epochs were also increased for 20 to 50. This change resulted in accuracy of 56% and loss closer to 0.77.

Results

The Details of the final model are below:

- There are in total 18 layers, out of which 8 are convolutional, and rest consists of dropout, pooling and dense layers.
- First convolution layer has stride 2 and rest has 1.
- Filter size used is 3x3.
- Number of filters in the convolutional layers are in increasing order, 32,32,64,64,128,128,256 and 256.
- Final layer is dense layer of size 3 equal to the number of size cervix type.

The final model achieved the accuracy about 56% on test set and cross entropy loss about 0.77 which is bit less than the benchmark 0.7. The prediction time of the final model is approx. 0.009 Second.

The performance of the model can be explained through the following Loss graph.

Figure 11

The final model's training results were following:

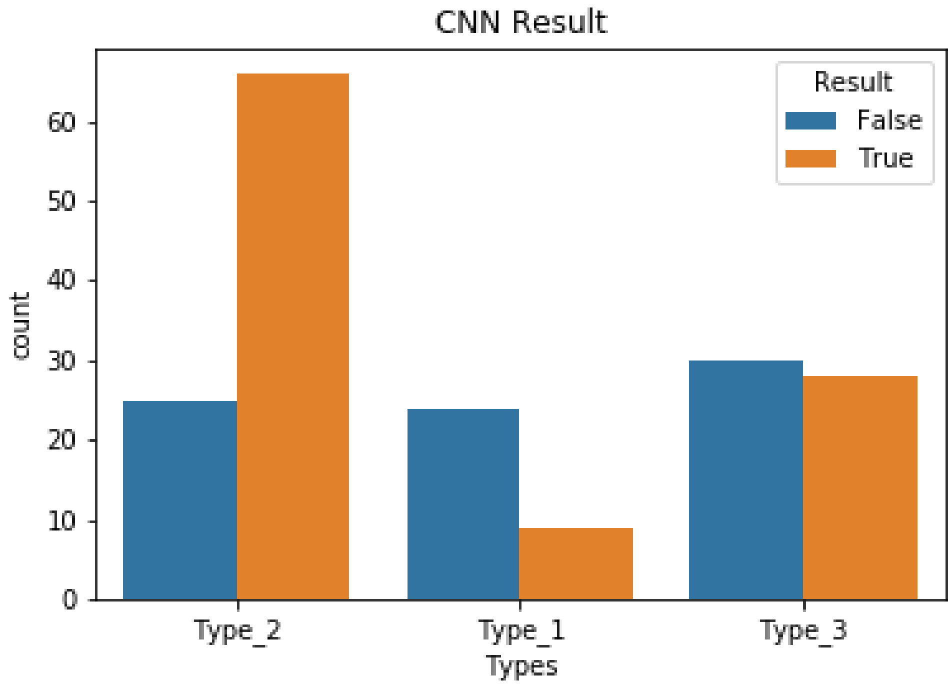


Figure 12

There could be many reasons as to why both models performed better on Type 2 but not on the other types. One of the major reasons could be number of samples of each type. Number of samples available for Type 2 is more than Type 1 and Type 3 combined.

On further analysis it was discovered that one way to observe the transformation zone of Cervix types is through the blood pattern.

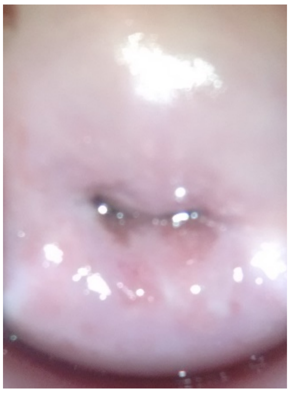
Type 1

Type 2

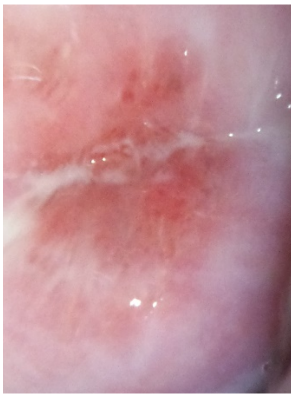
Type 3

From the above images it can be seen that, visibility of transformation zones decreases as we go from type 1 to type 3. The Model seem to do well when it finds similar blood patterns in the images but in cases such as below:

Type 2



Type 2



Type 1

Type 1

The blood patterns are not present but the transformation zones are visible, but the model classifies it as Type 3.

Conclusion

The whole process can be summarized in the following steps:

- Problem statement and data set were retrieved.
- As the problem statement was derived from the Kaggle competition, the benchmark was easy to decide.
- Data set was cleaned pre-processed.
- Coming up with the CNN architecture.
- Evaluating the model

I found Dataset cleaning and pre-processing was the tough part not because it was difficult, but it took a lot of time to manually frame and clean noisy images.

Interesting part in for me through the whole process was to fine tune the model to achieve better results.

For the further improvement, an App can be built which Health Care providers can use to figure the cervix type in real time.