Collaboration with

A.I Internship

Intern: Parikshit Saikia

Email: parikshitsaikia1619@gmail.com

Phone: 7636050549

Date: 29-08-2020

CONTENTS

1. Introduction

1.1 Overview

This is a project to automate the process detection of breast cancer using Deep techniques like Convolution Neural Network.
The goal is to make a complete web application where the user can upload the image of the mount slide and the machine learning model running in the backend will predict the output and send it to the html page which will be printed as output.

1.2 Purpose

Early detection of these diseases are very crucial. The purpose of the project is help the doctor to detect these deadly diseases, and to speed up this diagnostic process so that further procedures can be followed as soon as possible.

This project also aims to minimise the cost and labour required just to detect the presence of cancer in the slide.

2. Literature Survey

2.1 Existing Problem

Breast cancer is one of the main causes of cancer death worldwide.
Invasive ductal carcinoma (IDC), also known as infiltrating ductal carcinoma, is cancer that began growing in a milk duct and has invaded the fibrous or fatty tissue of the breast outside of the duct. IDC is the most common form of breast cancer, representing 80 percent of all breast cancer diagnoses.

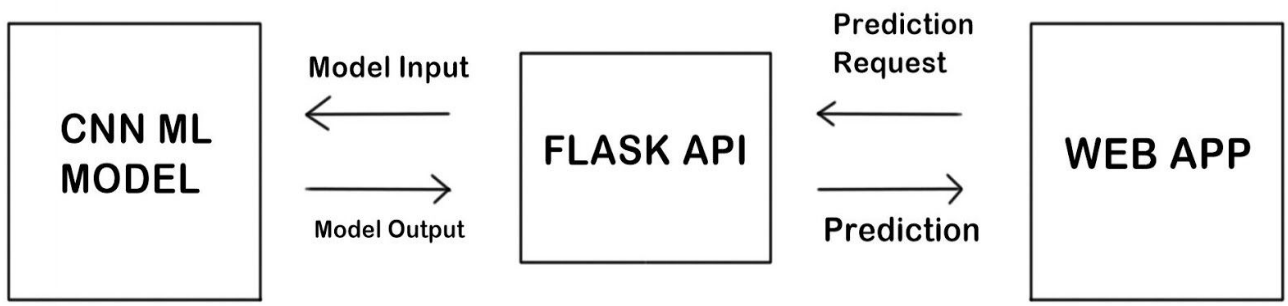
Early diagnostics significantly increases the chances of correct treatment and survival, but this process is tedious and often leads to a disagreement between pathologists. The chances of Human error are very high. Also the procedure is expensive due to the requirement of Laboratory equipment to do the diagnostics.

2.2 Proposed Solution

Computer-aided diagnosis systems showed potential for improving diagnostic accuracy.
Accurately identifying and categorizing breast cancer subtypes is an important clinical task, and Machine Learning techniques can be used to save time and reduce error. Techniques like Machine Learning and Image Processing can detect hidden features present in the mount slide, which can determine whether it has the risk of breast cancer or not.

3. Theoretical Analysis

3.1 Block Diagram



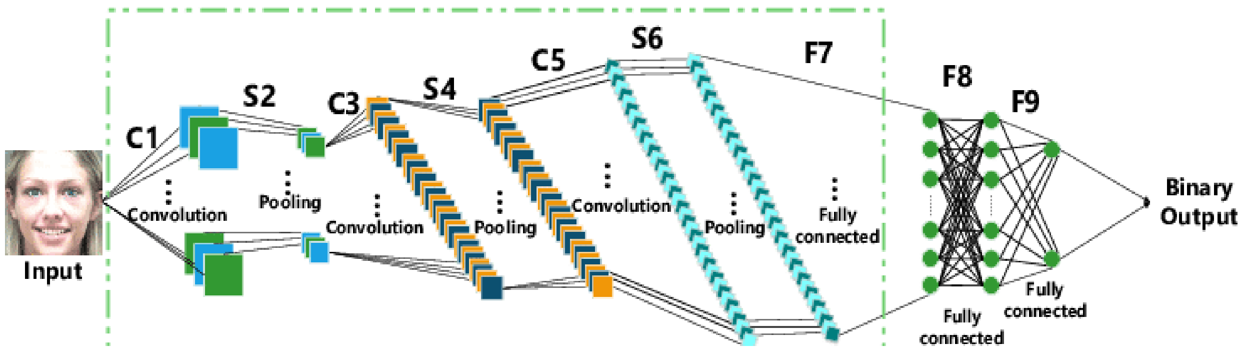
3.2 Software Designing

Since Breast Cancer detection falls in the category of image classification, A Convolution Neural Network Model will be best suitable for this.

To detect cancer from an image we need to extract important features from the image, due to the fact that these are medical images of tissue mount slides, extracting these features might be difficult.

That is why we are using 4 convolution layers in our model including batch normalisation and max pooling for each convolution layer.

This is an image of similar architecture we have used.



At the output we have only one node because our problem is of type binary classification.

- Cancer is detected (1)
- Cancer is not detected (0)

#code

```
# definning the model architecture
class SimpleCNN(nn.Module):
    def __init__(self):

        super(SimpleCNN, self).__init__()
        self.conv1 = nn.Conv2d(in_channels=3, out_channels=32, kernel_size=3, padding=2)
        self.conv2 = nn.Conv2d(in_channels=32, out_channels=64, kernel_size=3, padding=2)
        self.conv3 = nn.Conv2d(in_channels=64, out_channels=128, kernel_size=3, padding=2)
        self.conv4 = nn.Conv2d(in_channels=128, out_channels=256, kernel_size=3, padding=2)
        self.bn1 = nn.BatchNorm2d(32)
        self.bn2 = nn.BatchNorm2d(64)
        self.bn3 = nn.BatchNorm2d(128)
        self.bn4 = nn.BatchNorm2d(256)
        self.pool = nn.MaxPool2d(kernel_size=2, stride=2)
        self.avg = nn.AvgPool2d(8)
        self.fc = nn.Linear(256 * 1 * 1, 2) # !!!

    def forward(self, x):
        x = self.pool(F.leaky_relu(self.bn1(self.conv1(x)))) # first convolutional layer then batchnorm,
        x = self.pool(F.leaky_relu(self.bn2(self.conv2(x))))
        x = self.pool(F.leaky_relu(self.bn3(self.conv3(x))))
        x = self.pool(F.leaky_relu(self.bn4(self.conv4(x))))
        x = self.avg(x)
        #print(x.shape)
        x = x.view(-1, 256 * 1 * 1)
        x = self.fc(x)
        return x
```

4. Experimental Investigation

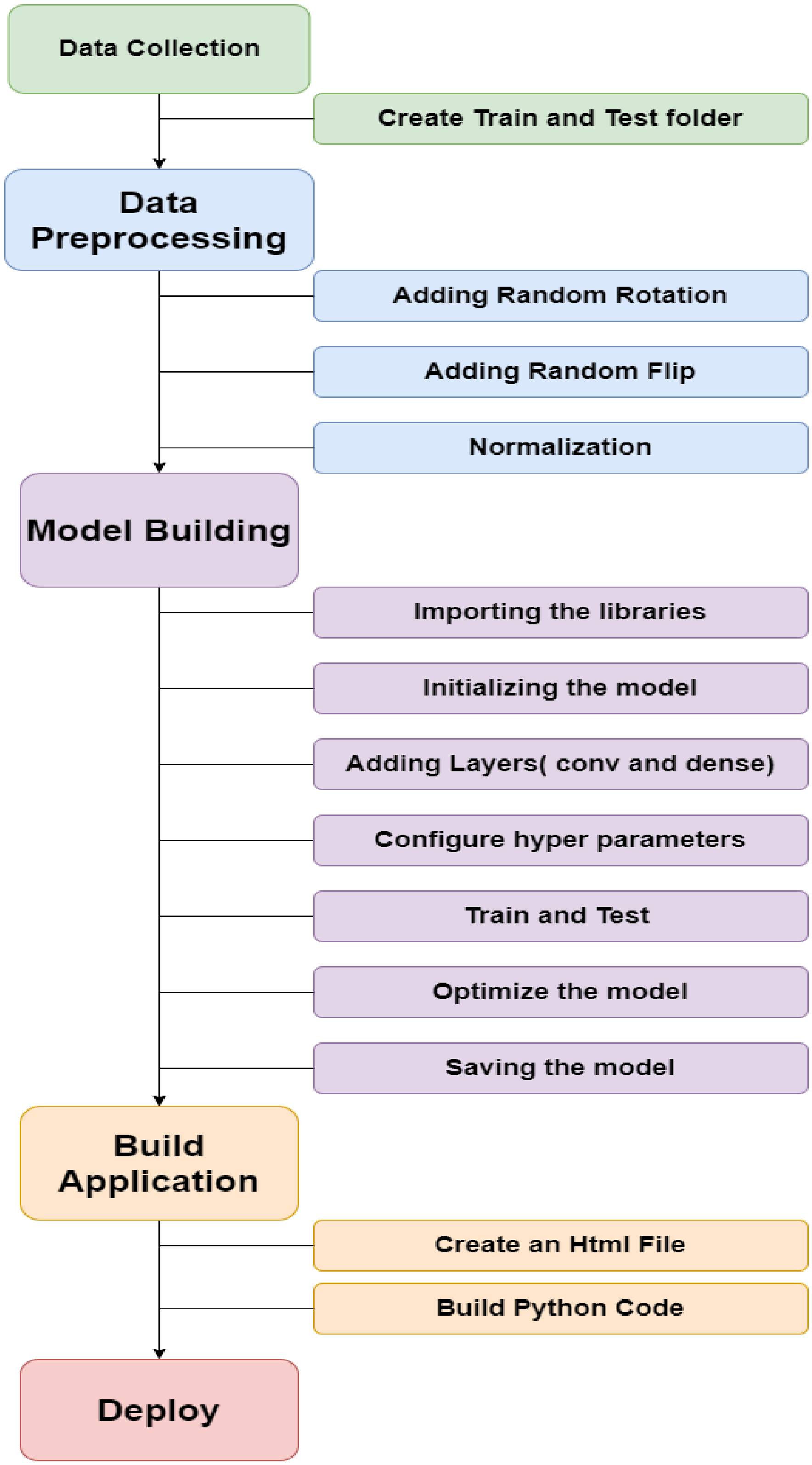
The main challenges occurred solving this problem was making a robust deep learning model , achieving high test accuracy, setting the hyper parameters.

- Robust Deep Learning Model:
    1. For this I had to do some research, and took inspiration from a state of the art CNN architecture paper.
    2. Since I had limited computing resources(GPU), I had to simplify the model architecture without compromising the performance.
    3. Adding more convolutional layers than 4 was not increasing the performance , So I ended up choosing 4 convolution layers.

- Achieving High test accuracy:
    - For this I transformed the images in the training set without random rotation , scaling and cropping. So that the model can generalise more.
    - Used adaptive learning rate to reach global minima.
    - Used Adam as optimizer, which increased the performance.
    - Used batch normalisation layer in the model architecture.

- Setting the hyper parameters:
    - Got some help with Hyper parameter tuning Research Papers.
    - Run different models with a different set of hyper parameters and select the set which gave best performance.

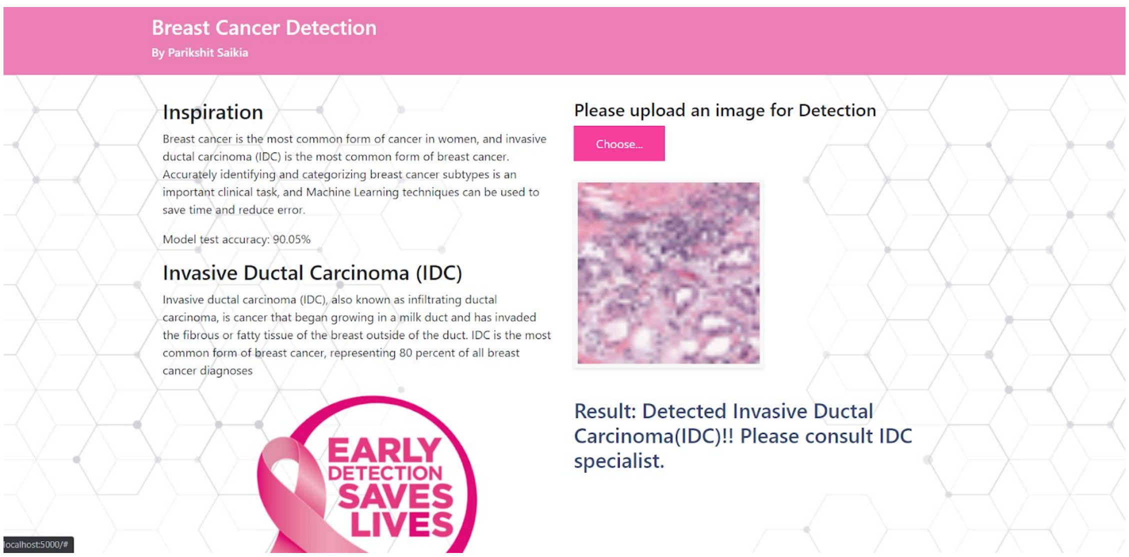
5. Flowchart

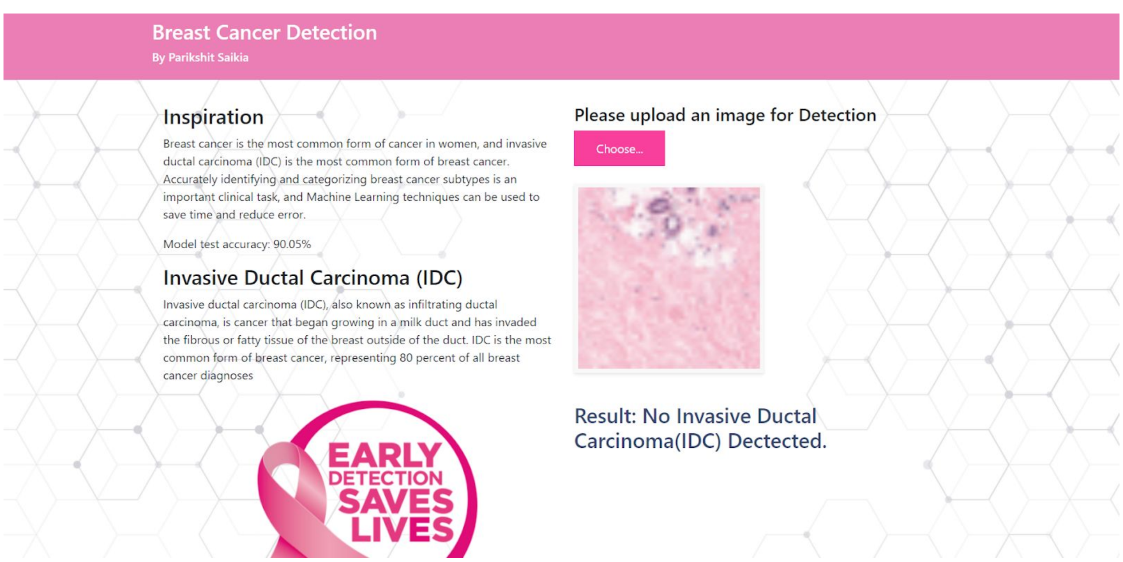

Data Collection
Create Train and Test folder
Data Preprocessing
Adding Random Rotation
Adding Random Flip
Normalization
Model Building
Importing the libraries
Initializing the model
Adding Layers(conv and dense)
Configure hyper parameters
Train and Test
Optimize the model
Saving the model
Build Application
Create an Html File
Build Python Code
Deploy

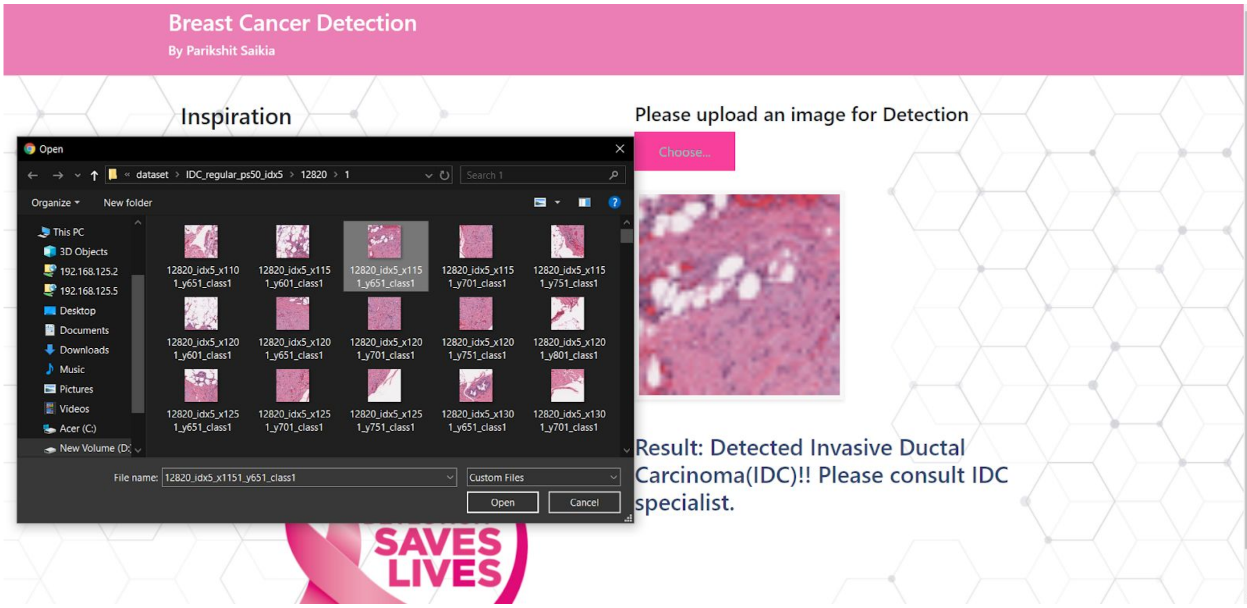
6. Result

After training and evaluating the model for a long time , I finally achieved 90% accuracy on the test data.
Also testing the model on the local host webpage it performed really well. It was classifying the mount slide correctly for most of the time.

Here are some images of the final result.







7. Advantages and Disadvantages

- Advantages:
  - Prediction can be done within no time.
  - Budget friendly.
  - No chance of Human error.
  - Reliable.
  - Open Source, easy to update the model for better results.

- Disadvantages:
  - Sometimes (rarely) predict inaccurate results.
  - Difficult to acquire unbiased dataset to train the model.
  - Training the model not to overfit.
  - Difficult to achieve high accuracy.

8. Application

This app can be used by doctors to consult with their patients. Since this app produces results very quickly , it will speed up the diagnostic procedure.
This app along with doctor’s experience can revolutionize the traditional cancer detection and treatment. It has the potential to decrease the mortality rate of cancer patients .

9. Conclusion

The whole project has been successfully completed, from collecting and preprocessing the dataset, building the model architecture, training and testing the model, making the webpage and finally deploying the model.
The model test accuracy is 90%, and it can correctly classify the images most of the time. The prediction is quite fast and finally the webpage is user friendly minimalist design.

10. Futurescope

For now the model is performing pretty well , but in medical classification we cannot take the risk of wrong prediction. So, it is better to make a more robust model .
We can change the model architecture to extract more features.
Collecting more unbiased datasets is always a way to improve.

Trying more hyper parameters to better is also an option.
And finally making the Html more user friendly and attractive, also hosting the web page.

11. Bibliography

Sources

https://www.mayoclinic.org/diseases-conditions/breast-cancer/symptoms-causes/syc-20352470

https://www.healthline.com/health/breast-cancer

https://www.kaggle.com/paultimothymooney/breast-histopathology-images

https://towardsdatascience.com/pytorch-basics-how-to-train-your-neural-net-intro-to-cnn-26a14c2ea29

https://www.w3schools.com/howto/howto_css_full_page.asp

12. Appendix

Source code:

https://github.com/SmartPracticeschool/llSPS-INT-3693-Deep-Learning-Techniques-for-Breast-Cancer-Risk-Prediction-using-Python/blob/master/Breast%20Cancer%20Risk%20Prediction.ipynb

UI screenshots are attached above.