

Mini Project -I Report
on
Breast Cancer Detection using CNN

by

Mihir Nikam
2019140042

Rushil Patel
2019140047

Ameya Ranade
2019140053

Guide Name:
Prof. Nikahat Mulla



Information Technology Department

Bharatiya Vidya Bhavan's

Sardar Patel Institute of Technology

Munshi Nagar, Andheri(W), Mumbai-400058

University of Mumbai

May 2021

Declaration

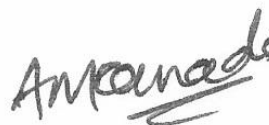
I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.



Mihir Nikam (2019140042)



Rushil Patel (2019140047)



Ameya Ranade (2019140053)

Date: 01/06/2021

Acknowledgements

We feel great pleasure in presenting the stage one report of our mini project titled 'Breast Cancer Detection using CNN'. We have channelized our best efforts towards a systematic approach to the project, keeping in mind the aim we need to achieve.

We are highly grateful to our project guide Prof. Nikahat Mulla, Department of Information Technology, Sardar Patel Institute of Technology (SPIT) for constant encouragement, effort and guidance. She has always been involved in discussing our topic at each phase to make sure that our approach was designed and carried out in an appropriate manner and that our conclusions were appropriate, given our results.

Mihir Nikam

Rushil Patel

Ameya Ranade

Abstract

Breast Cancer is the most popular and growing disease in the world. Breast Cancer is mostly found in women. Early detection is a way to control breast cancer. There are many cases that have dealt with breast cancer better due to early detection.

In India, 1 woman gets diagnosed with breast cancer every 4 minutes and 1 woman dies of breast cancer every 13 minutes, making it the most prevalent cancer among Indian women.

Women in India are generally diagnosed at a later, more advanced stage with poor prognosis.

Machine learning algorithms like decision trees, KNN, SVM, naïve bays etc. give good performance in their own field. A new technique has been developed to classify the tumor in breast cancer. This new developed technique is deep learning. Deep learning is used to overcome the drawbacks of machine learning.

Deep learning techniques are revolutionizing the field of medical image analysis and hence in this study, we proposed Convolutional Neural Networks (CNNs) for breast mass detection so as to minimize the overheads of manual analysis.

A deep learning technique that is mostly used in data science is Convolution neural network, Recurrent neural network, deep belief network etc. deep learning algorithms give better results as compared to machine learning. It extracts the best features of the images. In our research, CNN is used to classify the images. Basically our research is based on the images and CNN is the most popular technique to classify the images. In the present paper, reviews of all authors are conducted.

Table of Contents

1. Introduction	6
1.1. Problem Statement	6
1.2. Literature Survey/Market Survey	6
1.3. Scope and Objectives	7
1.4. Assumptions	7
1.5. Constraints	8
2. Proposed System	9
2.1. Architecture Diagram	9
2.2. Algorithms used	10
2.3. CNN Model Summary	13
3. Project Plan	15
4. Implementation	16
4.1. Step-wise Details	16
4.2. Tech Stack used	17
4.3. Results & Observations	17
4.4. Comparative analysis of our model	19
4.5. Module wise implementation screenshots	20
5. Conclusion and Further Work	22
References	23

1. Introduction

1.1 Problem Statement

Statement: Early breast cancer detection using CNN prediction model.

The most effective way to reduce breast cancer deaths is by detecting it earlier.

Diagnosis of breast cancer is done by classifying the tumor. Tumors can be either benign or malignant but only the latter is cancerous in nature.

Breast cancer is one of the main causes of cancer death worldwide. Early diagnostics significantly increases the chances of correct treatment and survival, but this process is tedious and often leads to a disagreement between pathologists. Computer-aided diagnosis systems showed potential for improving the diagnostic accuracy.

1.2 Literature Survey/Market Survey

1. Mandeep Rana, Pooja Chandorkar, Alishiba Dsouza and Nikahat Mulla, "***Breast Cancer Diagnosis and Recurrence Prediction Using Machine Learning Techniques***", Paper. eISSN: 2319-1163 | pISSN: 2321-7308, 2015.

Summary:

A subtle comparison of different ML techniques and their efficiency in predicting breast cancer is done. Most of them perform with decent accuracy, SVM and k-NN being the more significant stand outs.

2. Nrea, Simon & Gezahegn, Yaacob & Sinamo, Abiot & Hagos, Gebrekirstos. "***Breast Cancer Detection Using Convolutional Neural Networks***", 2020

Summary:

This paper describes a CNN based approach for detection of mass regions and classifies them into benign or malignant abnormalities in mammogram(MG) images at once. It does so, by enhancing the quality of image, RELU layer activation, batch normalization, max pooling layer and dropout. Accuracy of up to 91.86% is achieved with this system.

3. [1]Rakhlin, A., Shvets, A., Iglovikov, V., and Kalinin, A. A., ***“Deep Convolutional Neural Networks for Breast Cancer Histology Image Analysis”***, 2018.

Summary:

This paper proposes histology image analysis using deep CNN for Feature Extraction and Gradient Boosted Trees. For image based predictions, feature extraction is heavily used whichever the technology in question. The dataset used has been stained and cropped into pieces. After which, the cropped pieces are encoded.

1.3 Scope and Objectives

Scope:

Building a prediction model for breast cancer detection using CNN. Once the model is built, we will integrate it into a web application. Once the final product is ready, we plan to apply it on real-time data.

Objectives:

- Prevention is better than cure. In case of breast cancer, all the more essential since there is a risk of cancer spreading to other parts of body
- The physical methods for analysis of Breast cancer takes a lot of time.
- So, our main objective is to reduce the time it takes to detect whether a patient has Breast cancer or not.

1.4 Assumptions

- 1.1.1. The input image should be an RGB image of the biopsy tissue.
- 1.1.2. The input dimension of the image should be (50px*50px*50px)
- 1.1.3. The image which is to be uploaded must be well lit and clear.
- 1.1.4. Since this is a prediction model based on already calculated probabilities and chances, accuracy may be hampered for some test cases.

1.5 Constraints

- 1.1.5. We have used histopathology image dataset. The prediction is solely based on the image uploaded. It does not take any other details into consideration.
- 1.1.6. This model is not suitable for all types of people since the dataset used is of a particular ethnicity.
- 1.1.7. Since this is a prediction model based on already calculated probabilities and chances, accuracy may be hampered for some test cases.
- 1.1.8. This dataset only classifies whether the tissue is benign or malignant. We do not take the patient's family history into consideration. This may cause the data to be vulnerable.

2. Proposed System

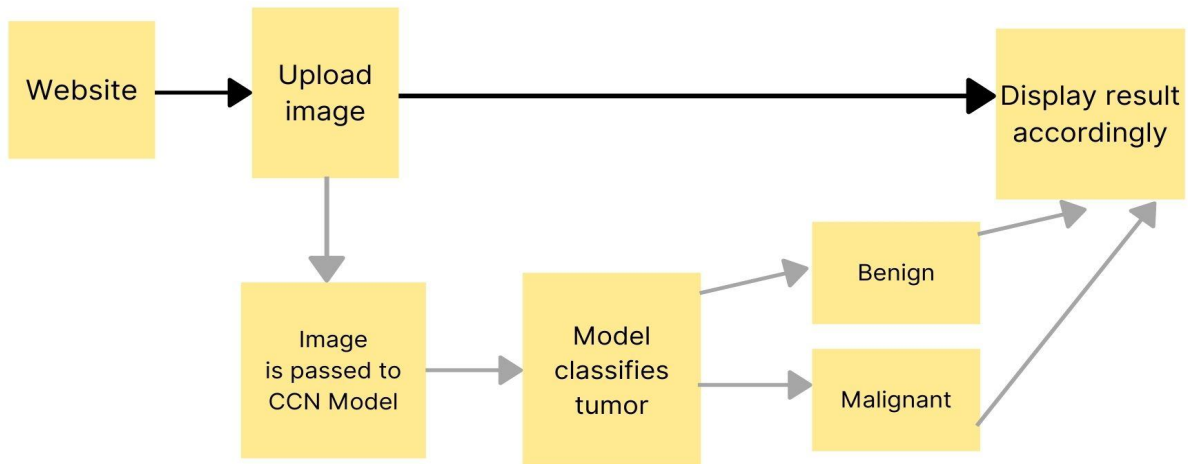


Figure 2.1, Proposed System Diagram

- 2.1.1. **Website:** Interface to upload biopsy image for prediction.
- 2.1.2. **Upload Image:** Upload image from your local device.
- 2.1.3. **Model Works Behind the Scenes:** Once the image is uploaded on StreamLit, the image undergoes feature extraction. Post this, the image passes through convolutional layers and the pooling layers to successfully classify the tumor into its appropriate class.
- 2.1.4. **Classification of Tumor:** Tumor is classified in terms of 0 and 1. 0 denotes benign whereas 1 denotes malignant.
- 2.1.5. **Display Result:** According to the class of tumor, the website will display the result.
 - If the class of tumor is benign, i.e. the patient is safe. The result is displayed with a green background.
 - If the class of tumor is malignant, i.e. the tissue is cancerous. The result is displayed with a red background.

2.2 Algorithms used

Algorithm implementation : Convolutional Neural Networks (CNNs)

CNN: CNNs are applied to explore patterns in an image. This is done by convoluting over an image and looking for patterns. The network can detect lines and corners in the few front layers of CNNs. Via our neural net, however, we can then transfer these patterns down and begin to identify more complex characteristics as we get deeper. This property ensures that CNNs are very effective at detecting objects in images. The proposed system uses CNNs to detect breast cancer from breast tissue images.

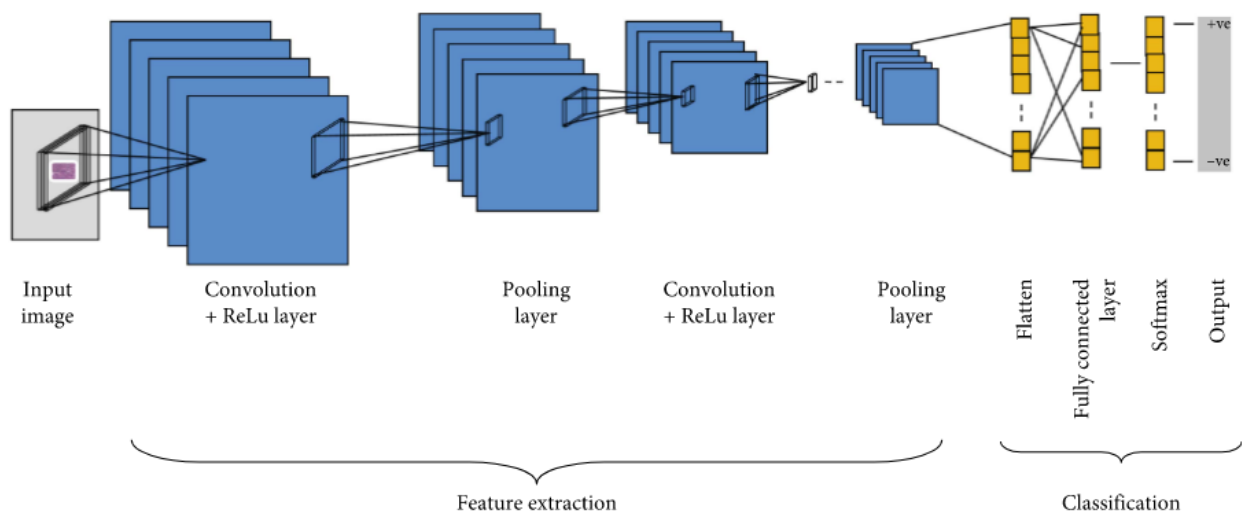


Figure 2.2, Image of CNN model with different layers

- **Characteristics of our model**

- we have used (3,3) size filter
- we have not padded the input images i.e padding = "same"
- we have used a max_pooling layer of size (2,2)
- we have used "Relu" as the activation function
- we have also implemented Batch_normalization for better efficiency
- we have used dropout optimisation to avoid overfitting of our model

- **Activation Function : Relu**

An activation function in a neural network defines how the weighted sum of the input is transformed into an output from a node or nodes in a layer of the network.

The rectified linear activation function or ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero.

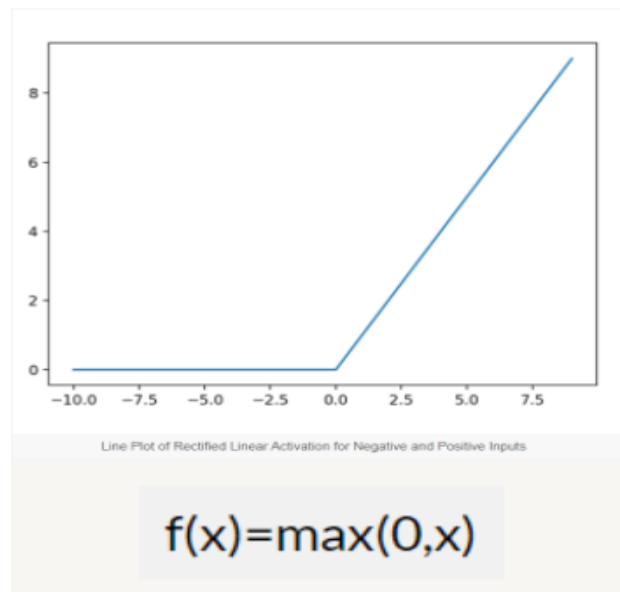


Figure 2.3: Activation function used as in Research Paper 2

- **Batch Normalization**

Batch normalization is a layer that allows every layer of the network to do learning more independently. It is used to normalize the output of the previous layers. Using batch normalization learning becomes efficient also it can be used as regularization to avoid over-fitting of the model. The layer is added to the sequential model to standardize the input or the outputs. It can be used at several points in between the layers of the model.

(Data augmentation from Research Paper 3)

- **Dropout Regularization**

A fully connected layer occupies most of the parameters, and hence, neurons develop co-dependency amongst each other during training which curbs the individual power of each neuron leading to over-fitting of training data.

(Dropout Layer used as in Research Paper 2)

Our CNN model for Breast Cancer Detection

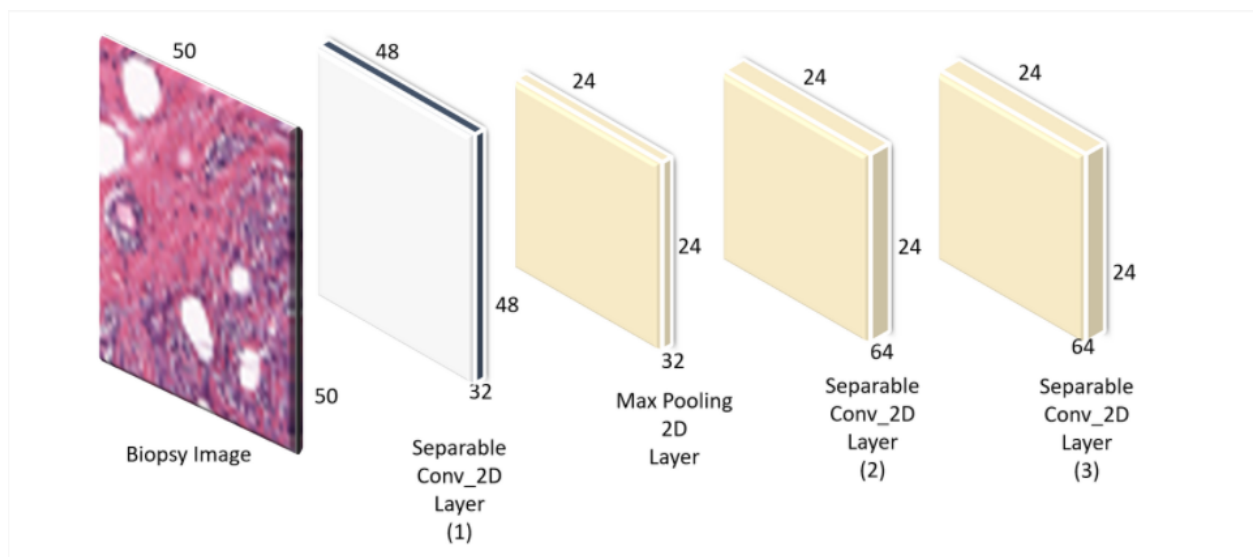


Figure 2.4

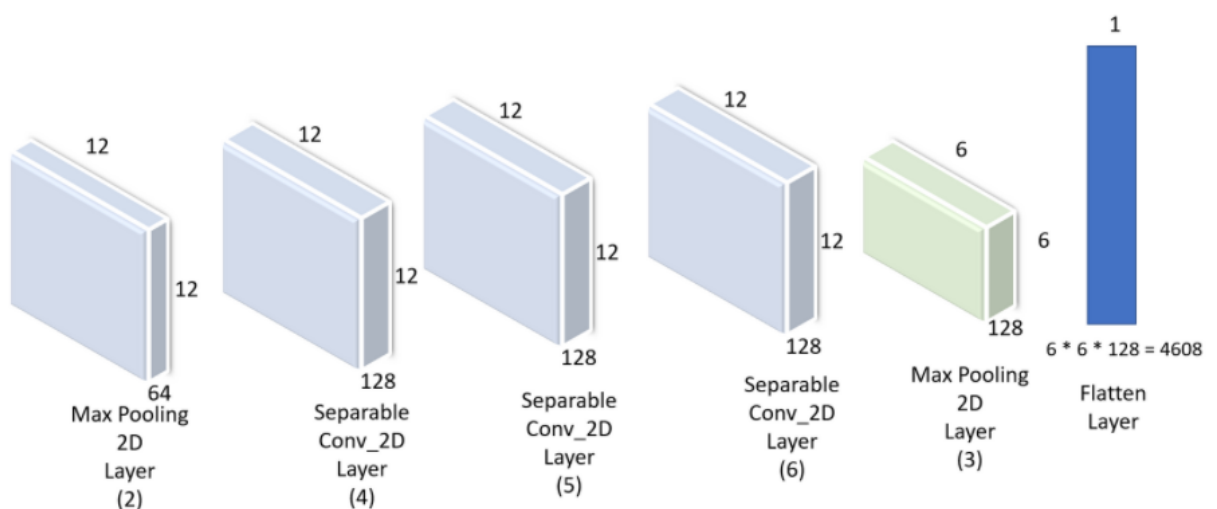


Figure 2.5, CNN Architecture

2.3 CNN Model Summary

Layer (type)	Output Shape	Param #
separable_conv2d_7 (Separabl	(None, 50, 50, 32)	155
activation_9 (Activation)	(None, 50, 50, 32)	0
batch_normalization_8 (Batch	(None, 50, 50, 32)	128
max_pooling2d_4 (MaxPooling2	(None, 25, 25, 32)	0
dropout_5 (Dropout)	(None, 25, 25, 32)	0
separable_conv2d_8 (Separabl	(None, 25, 25, 64)	2400
activation_10 (Activation)	(None, 25, 25, 64)	0
batch_normalization_9 (Batch	(None, 25, 25, 64)	256
separable_conv2d_9 (Separabl	(None, 25, 25, 64)	4736
activation_11 (Activation)	(None, 25, 25, 64)	0
batch_normalization_10 (Batc	(None, 25, 25, 64)	256
max_pooling2d_5 (MaxPooling2	(None, 12, 12, 64)	0
dropout_6 (Dropout)	(None, 12, 12, 64)	0
separable_conv2d_10 (Separab	(None, 12, 12, 128)	8896
activation_12 (Activation)	(None, 12, 12, 128)	0
batch_normalization_11 (Batc	(None, 12, 12, 128)	512
separable_conv2d_11 (Separab	(None, 12, 12, 128)	17664
activation_13 (Activation)	(None, 12, 12, 128)	0
batch_normalization_12 (Batc	(None, 12, 12, 128)	512
separable_conv2d_12 (Separab	(None, 12, 12, 128)	17664

Figure 2.6

separable_conv2d_12 (Separab	(None, 12, 12, 128)	17664
activation_14 (Activation)	(None, 12, 12, 128)	0
batch_normalization_13 (Batc	(None, 12, 12, 128)	512
max_pooling2d_6 (MaxPooling2	(None, 6, 6, 128)	0
dropout_7 (Dropout)	(None, 6, 6, 128)	0
flatten_2 (Flatten)	(None, 4608)	0
dense_3 (Dense)	(None, 256)	1179904
activation_15 (Activation)	(None, 256)	0
batch_normalization_14 (Batc	(None, 256)	1024
dropout_8 (Dropout)	(None, 256)	0
dense_4 (Dense)	(None, 2)	514
activation_16 (Activation)	(None, 2)	0
=====		
Total params: 1,235,133		
Trainable params: 1,233,533		
Non-trainable params: 1,600		

Figure 2.7, Model.summary() Screenshot

3. Project Plan

The whole project can be roughly divided into five sections:

- a. Data collection and Data formatting.
- b. Data splitting - Test and Training Set.
- c. Building a CNN model using the collected data.
- d. Training and testing data repeatedly.
- e. Integrating the final model into a webpage.

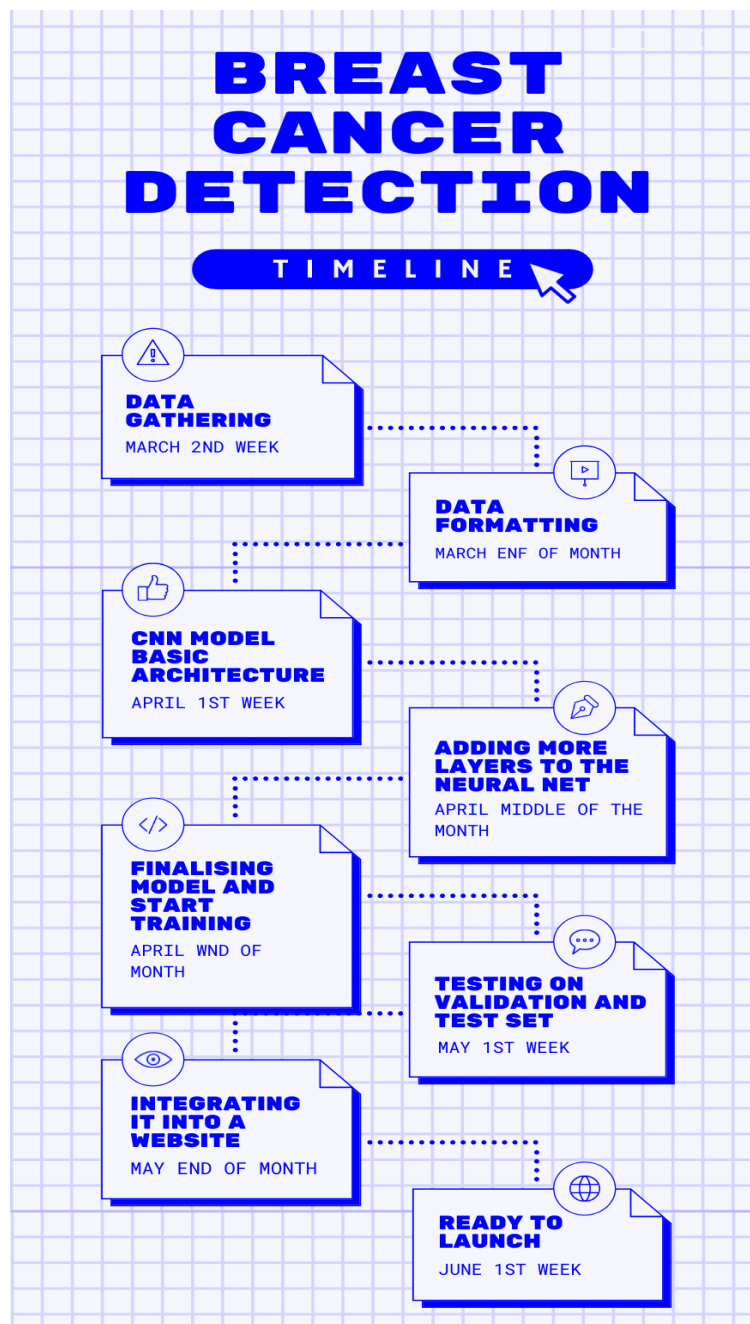


Figure 3.1, Project Timeline

4.Implementation

Details of 100% of implementation completed:

4.1 Step-Wise Details:

1. Dataset used is The Breast Histopathology Image Dataset from Kaggle. The original dataset consisted of 162 whole mount slide images of Breast Cancer (BCa) specimens scanned at 40x. From that, 277,524 patches of size 50 x 50 were extracted (198,738 IDC negative and 78,786 IDC positive). Each patch's file name is of the format: *uxXyYclassC.png* — > *example 10253idx5x1351y1101class0.png* . Where *u* is the patient ID (10253idx5), *X* is the x-coordinate of where this patch was cropped from, *Y* is the y-coordinate of where this patch was cropped from, and *C* indicates the class where 0 is non-IDC and 1 is IDC.
2. We wrote a code in a separate file named '**build_dataset.py**' which first shuffles all the images from the dataset using a random function, to get rid of any biases if any. This shuffled data is then divided into 3 parts that are Training, Validation and Testing Sets in a ratio of 0.8 : 0.1 : 0.1 respectively.
3. The CNN model which is the backbone of the whole project is written in a file named '**cancernet.py**'. In this we have made a convolutional neural net, the details of it are explained above in section 2.2
4. '**config.py**' is used to configure the CNN model. It is the initialisation of the model, and it tells the model where the dataset split into 3 different sets are located in the memory.
5. Lastly, after performing data augmentation, data splitting and building CNN model we have trained it on the data, the code for it is written in '**train_model.py**'. After training the model we have tested it against the validation and testing data and have summarized that using a confusion matrix. Finally, we have saved the model as '**my_model.h5**' file so it can be used to make further predictions.
6. '**app.py**' file contains the code where it takes in the user uploaded image and processes it using the earlier saved model to make predictions.
7. All this is integrated into a website using 'Streamlit Framework' which renders it with the frontend.

4.2 Tech Stack Used:

- ❖ Frontend:
 - **Streamlit:** Streamlit turns data scripts into shareable web apps in minutes. It's an open-source app framework for Machine Learning and Data Science teams.
- ❖ CNN Model:
 - Python
 - **TensorFlow:** TensorFlow is an end-to-end open source platform for machine learning.
- APIs used:
 - ❖ TensorFlow adopted **Keras** as the high-level API

4.3 Results & Observations

Graphs/comparative tables/ observations from results:

- ❖ We have plotted a line graph of training loss and accuracy vs number of epochs using matplotlib which shows the models performance as it is being trained.
- ❖ **Inference from the graph:**
 - Training loss decreases gradually as the model is being trained

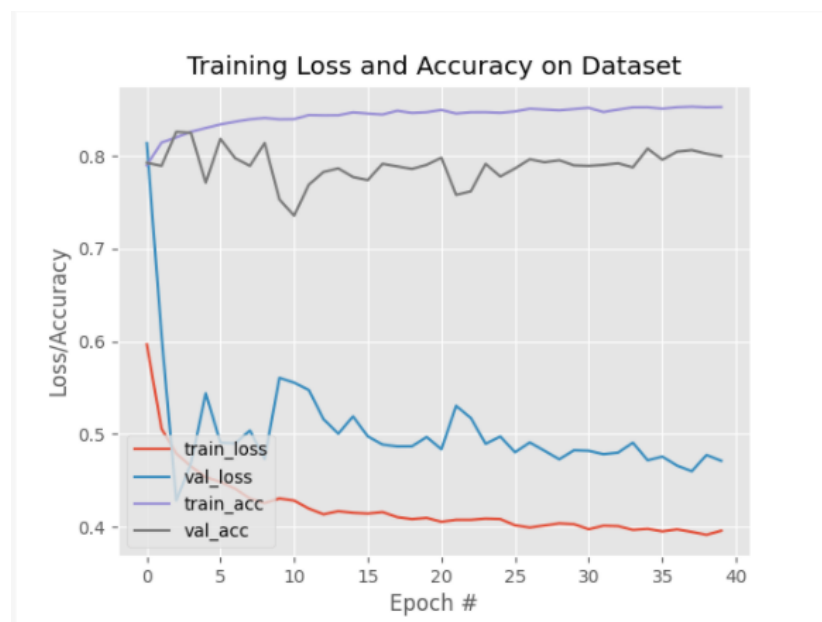
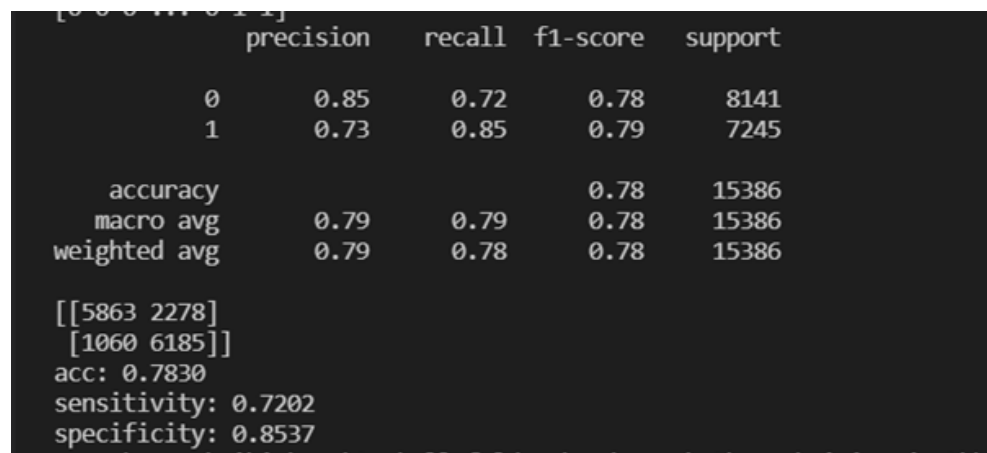


Figure 4.1, Result Graph

- Training accuracy is also increasing gradually and doesn't show any huge deviations which is a sign that the model isn't overfitting the training set
 - validation loss decreases gradually as the model is being trained
 - validation accuracy also flattens out at around 80% as the number of epochs increase
- ❖ Along with this we also take inferences from the Confusion matrix.



	precision	recall	f1-score	support
0	0.85	0.72	0.78	8141
1	0.73	0.85	0.79	7245
accuracy			0.78	15386
macro avg	0.79	0.79	0.78	15386
weighted avg	0.79	0.78	0.78	15386

[[5863 2278]
[1060 6185]]
acc: 0.7830
sensitivity: 0.7202
specificity: 0.8537

Figure 4.2, Confusion Matrix

- ❖ The F1 score is a good measure of a model's accuracy, higher the F1 score better the model performs
- ❖ Our model's final accuracy is around 79% with an F1 score of 0.78

4.4 Comparative analysis of model(CNN) vs other machine learning models

ML technique	Training Accuracy (%)	Validation / Test Accuracy(%)
SVM-linear	74	68
SVM-RBF-static “C” parameter	100	68
SVM-RBFdynamic “C” parameter	100	64
Logistic regression generalized	82	68
Logistic regression regularized	80	72
k-NN-Euclidean	100	72
k-NN-Manhattan	100	70
Naïve Bayes normal	81.33	66
Naïve Bayes kernel	81.33	67
CNN (our model)	91	79

Figure 4.3, Comparison Table

(This table entries are taken from Research paper 1)

This table shows a comprehensive comparison between CNN model vs other ML techniques used to detect Breast Cancer.

K-NN and SVM are also one of the most sought out techniques which are used for classification problems and have performed decently enough.

But the deep learning CNN models out performs these algorithms as CNN is a more powerful method for image classification(image dataset). It learns directly from the image itself and hence is able to infer more acute information. But on the other hand CNN involves continuous forward propagation and backward propagation while learning, so it takes a lot of time to train.

4.5 Module-wise Implementation Screenshots

Home page:

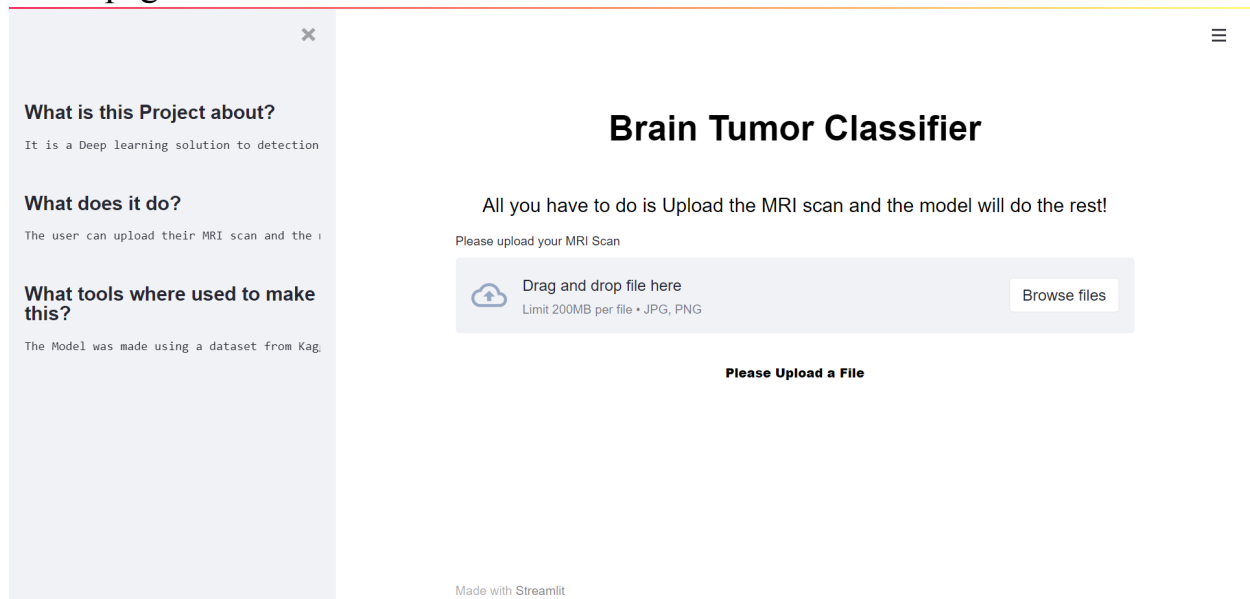


Figure 4.4 Home Page

Case 1: Benign Tissue

Result:

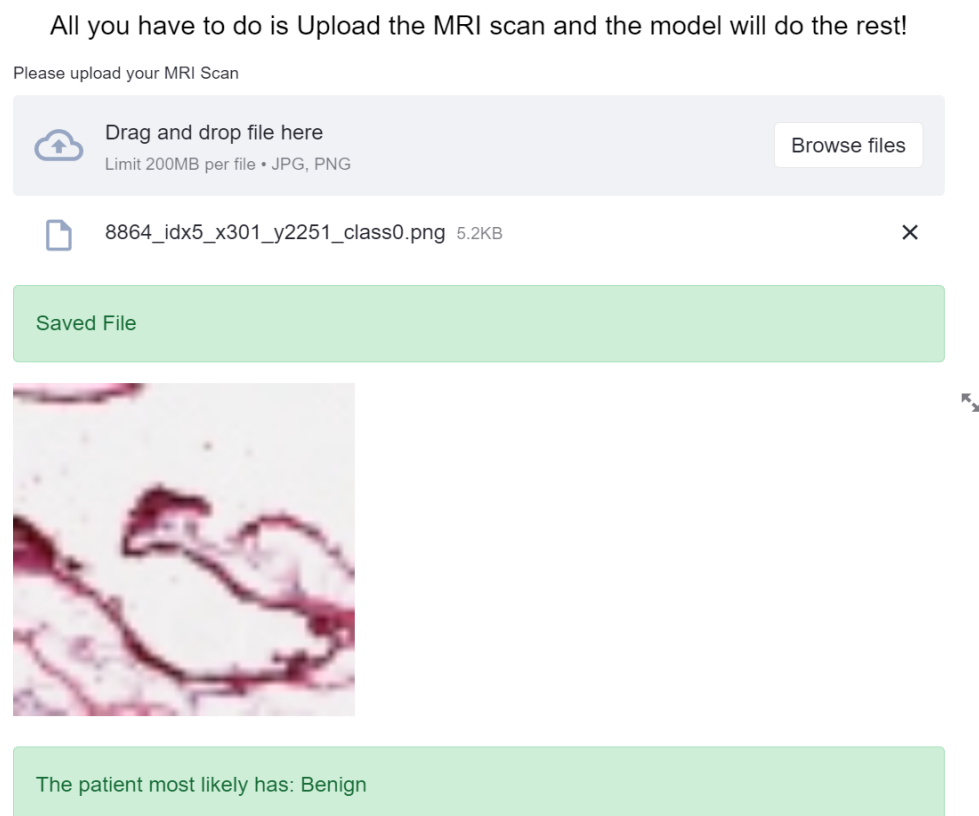


Figure 4.5 Negative Sample

Case 2: Malignant Tissue

Result:

Please upload your MRI Scan



Drag and drop file here

Limit 200MB per file • JPG, PNG

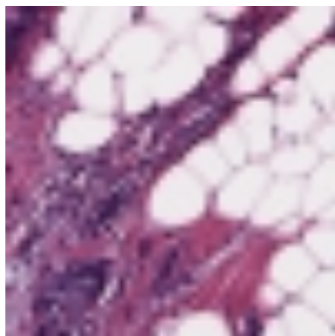
Browse files



8864_idx5_x2251_y1951_class1.png 6.3KB



Saved File



The patient most likely has: Malignant

Figure 4.6 Positive Sample

5. Conclusion and Further Work

We can see above that the most effective way of reducing breast cancer deaths is by detecting it earlier. Hence, we predict if it is benign or malignant using the machine learning model on the histopathology images. The accuracy achieved is roughly 80 percent and the f1-score is 0.8. This is further integrated into a website that can be used by users all over the world to get their samples tested and hence it is of great use to humanity. Our main motive behind taking up this project was to contribute to saving patients from cancer and it's traumatic effect in one's life.

This project as of now is being implemented on a small scale and won't be able to reach out to all the aspects of Breast Cancer Detection. But we plan to make this model more robust further down the lane by collecting more data and optimising it more and more such that it becomes efficient enough to someday really be used in the real time medical profession. As more and more people use the websites we plan to use those uploaded images of the biopsy and include them into our data(only by user's consent). This will enable us to continuously gather more and divergent types of data, which will improve the scope of our model.

References

Write all referred material (research papers/links/tutorials/blogs/APIs used) in IEEE standard reference style.

Research papers:

1. Mandeep Rana, Pooja Chandorkar, Alishiba Dsouza and Nikahat Mulla, ***“Breast Cancer Diagnosis and Recurrence Prediction Using Machine Learning Techniques”***, Paper. eISSN: 2319-1163 | pISSN: 2321-7308, 2015.
2. Nrea, Simon & Gezahegn, Yaacob & Sinamo, Abiot & Hagos, Gebrekirstos. ***"Breast Cancer Detection Using Convolutional Neural Networks"***, 2020
3. [1]Rakhlin, A., Shvets, A., Iglovikov, V., and Kalinin, A. A., ***“Deep Convolutional Neural Networks for Breast Cancer Histology Image Analysis”***, 2018.

Tutorials/Documentation:

1. “StreamLit Documentation” streamlit.io[Online]. Available: <https://docs.streamlit.io/en/stable/index.html>. [Accessed 02, May, 2021].
2. “TensorFlow API Docs”. <https://www.tensorflow.org>[Online]. Available: https://www.tensorflow.org/api_docs. [Accessed 19, Feb, 2021].