Deep Learning project

Breast Cancer detection using deep learning

Himanshu Rahi (B20BB014)
Romit Kalal (B20BB032)
Shovit Sharma (B20BB038)

Introduction

Breast cancer is a significant cause of death among women globally. Early detection of breast cancer is crucial for effective treatment and can significantly improve survival rates. However, traditional methods of detecting breast cancer , such as mammography, have limitations in terms of accuracy and sensitivity.

Therefore, there is a need to develop more effective methods of breast cancer detection. The key challenge against its detection is how to classify tumors into malignant (cancerous) or benign(non cancerous).

The objective of this project is to develop a neural network-based classification model to accurately predict the presence or absence of breast cancer. The model will be trained using a dataset of patient information and clinical data. The model will then be tested on a separate dataset to evaluate its accuracy and performance.

Datasets

The Datasets used are " **Wisconsin (Diagnostic)** "and the " **BreakHis** " Breast Cancer Histopathological Database.

Wisconsin Dataset

In this the features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. This describes the characteristics of the cell nuclei which were present in the image.

Attribute Information :-

1) ID number

2) Diagnosis (M = malignant, B = benign)

Ten real-valued features are computed for each cell nucleus:

1. Ten real-valued features are computed for each cell nucleus:
2. Radius (mean of distances from center to points on the perimeter)
3. Texture (standard deviation of gray-scale values)
4. Perimeter
5. Area
6. Smoothness (local variation in radius lengths)
7. Compactness (perimeter^2 / area - 1.0)
8. Concavity (severity of concave portions of the contour)
9. Concave points (number of concave portions of the contour)
10. Symmetry
11. Fractal dimension ("coastline approximation" - 1)

For each image the “**mean**”, “**standard error**” and "**worst**" or largest (mean of the three largest values) of these features were computed , which results in 30 features.For example field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius. All feature values are recorded with four significant digits.

BreakHis Dataset

The BreaKHis database contains microscopic biopsy images of benign and malignant breast tumors.It is composed of around 9,109 microscopic images of breast tumor tissue collected from 82 patients using different magnifying factors (40X, 100X, 200X, and 400X). But for our model we had used (400X images only).

Model 1

It contains three FC (linear) layers, with input neurons counted as 30 , 16 hidden neurons, 4 hidden neurons and 1 output neuron.The activation function used here is “Sigmoid” after each fully connected layer.

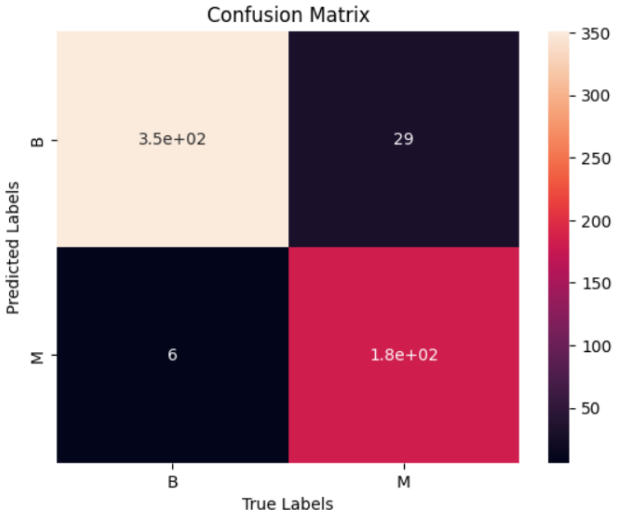
The FC1 layer is defined as a fully connected layer with 30 input neurons and 16 output neurons. Similarly for other FC layers with 16 and 4 acting as input neurons and 4 and 1 acting as output neurons respectively. The input x is passed through the FC1 layer and the result is passed through the sigmoid activation function. The output of the sigmoid function is passed through the FC2 layer, and again through the sigmoid function. Finally, the output of the sigmoid function is passed through the fc3 layer and the final output y_pred is obtained. The sigmoid function is applied again to ensure that the output is between 0 and 1, which is useful for binary classification tasks.

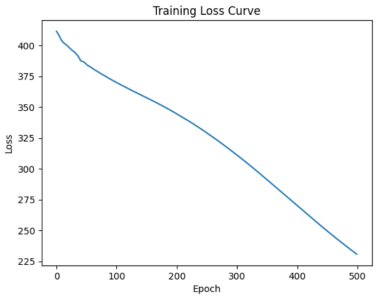
Model 2

It is a Convolutional Neural Network (CNN) model using the PyTorch nn module.This model contains three Convolutional Layers, each followed by activation function (ReLu) and a Max Pooling layer. For feature extraction the convolution layers are used from the input image while the max pooling layers are used to downsample the output of the Convolution layers. The output is flattened after the 3rd convolution layer using the nn.Flatten() layer, which converts the multidimensional tensor to a one-dimensional tensor. This flattened tensor is then passed through two FC layerswith ReLU activation functions, where the final output has two neurons representing the two classes.In summary, the model takes a 3-channel input image (colored breast tissue image), applies three convolutional layers with ReLU activations and max pooling, then flattens the output and passes it through two fully connected layers with ReLU activations to output the final classification for the input image.

Result

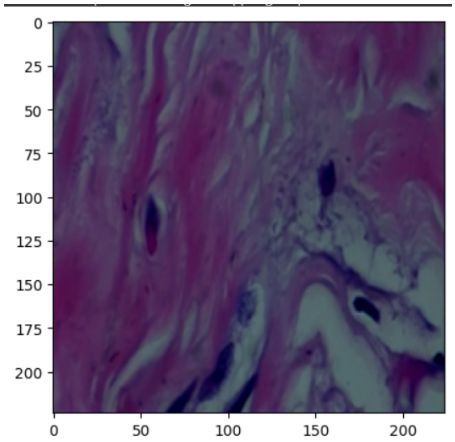
Model1





	precision	recall	f1-score	support
False	0.98	0.92	0.95	380
True	0.86	0.97	0.91	189
accuracy			0.94	569
macro avg	0.92	0.95	0.93	569
weighted avg	0.94	0.94	0.94	569

Model2 :



Epoch [1/10], Train Loss: 0.0616, Test Loss: 0.9205, Train Accuracy: 0.98%, Test Accuracy: 0.81%
Epoch [2/10], Train Loss: 0.0419, Test Loss: 0.8490, Train Accuracy: 0.99%, Test Accuracy: 0.82%
Epoch [3/10], Train Loss: 0.0269, Test Loss: 1.0245, Train Accuracy: 0.99%, Test Accuracy: 0.84%
Epoch [4/10], Train Loss: 0.0052, Test Loss: 1.1475, Train Accuracy: 1.00%, Test Accuracy: 0.83%
Epoch [5/10], Train Loss: 0.0016, Test Loss: 1.1784, Train Accuracy: 1.00%, Test Accuracy: 0.83%
Epoch [6/10], Train Loss: 0.0023, Test Loss: 1.2972, Train Accuracy: 1.00%, Test Accuracy: 0.82%
Epoch [7/10], Train Loss: 0.0007, Test Loss: 1.3661, Train Accuracy: 1.00%, Test Accuracy: 0.83%
Epoch [8/10], Train Loss: 0.0004, Test Loss: 1.4018, Train Accuracy: 1.00%, Test Accuracy: 0.83%

Epoch [9/10], Train Loss: 0.0003, Test Loss: 1.4476, Train Accuracy: 1.00%, Test Accuracy: 0.82%
Epoch [10/10], Train Loss: 0.0002, Test Loss: 1.4849, Train Accuracy: 1.00%, Test Accuracy: 0.83%

Conclusion

Model 2 is better suited for breast cancer detection between M (malignant) and B (benign) compared to the model1 .

The Model 2 uses (CNNs) which are specifically designed to work with image data. It consists of three convolutional layers, each with an increasing number of filters and followed by ReLU activation functions and max-pooling layers. This will help us to extract features from the input image and reduce its dimensions. The output from the convolutional layers is then flattened and passed through two fully connected layers with ReLU activation functions and a final linear layer with 2 outputs (representing the M and B classes). This architecture is commonly used for image classification tasks and has been shown to perform well in various benchmark datasets.

On the other hand, the first model is a simple feedforward neural network (FFNN) with three fully connected layers and sigmoid activation functions. It takes a 30-dimensional input and outputs a single value between 0 and 1, which may not be sufficient to differentiate between the M and B classes. While the sigmoid function can be used to represent binary classification problems, it is not optimized for image data and may not perform well compared to CNNs.