Breast Cancer Detection

- Aditya Agarwal
- Himani Thakker
- Pushkar

Overview

- Breast cancer is the second leading cause of death in women. Breast cancer mortality is reduced by early identification and treatment.

  Implementation of this project can help in the following ways:

  1. Help physicians for early detection to maximize patients' survival rate.

  2. Minimize the number of "untrained eyes" that is wrong interpretations and increase the accuracy of screening.

  3. Prevent late treatments as well as unnecessary treatments in case of false positives.

  4. One would be able to overcome the dependency of pathologist in the places where no experts are available.

Dataset

The dataset to be used in this project is put together by Mendeley Data. It is free to use and open source. The dataset contains mammography with benign and malignant masses.

1. INbreast -

Total Images : 7632

Train – 5494

Test – 1527

Validation - 611

2. MIAS (Mammographic Imaging Analysis society) -

Total Images : 3816

Train – 2746

Test - 764

Validation - 306

Image Size : 227 * 227

Approach

1. Image transformations: Rescaled the image from 225 to 0-1 scale

2. Partitioned the dataset into training, testing and validation dataset

3. Baseline CNN model –

- A) 2 layers with 32 kernels each and size of (3,3) respectively,
- B) Used Pooling layer after each Convolutional layer
- C) To bring the output of the neural network to a probability distribution over predicting classes, we have used Softmax function

4. Advanced CNN model –

A) Used two additional layers with 64 and 128 kernels respectively with size as (3,3)

5. We have used Categorical cross-entropy as our loss function to distinguish the probability distribution of benign and malignant from each other.

6. Adam optimizer to update the weights and learning rate accordingly.

7. To evaluate the model performance, we have used "Accuracy" as our performance metric.

CNN Model Architecture

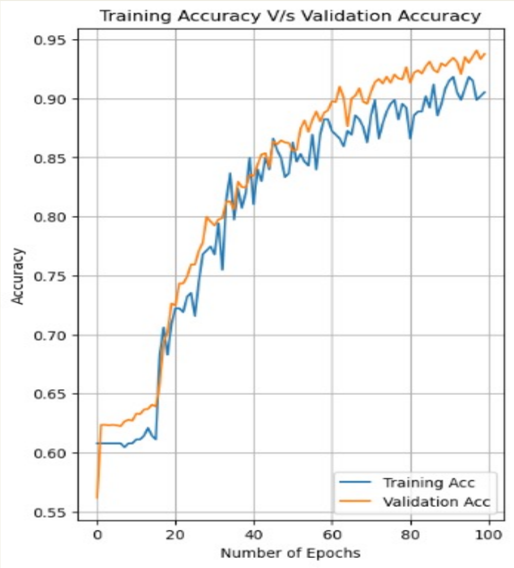
Baseline CNN Model

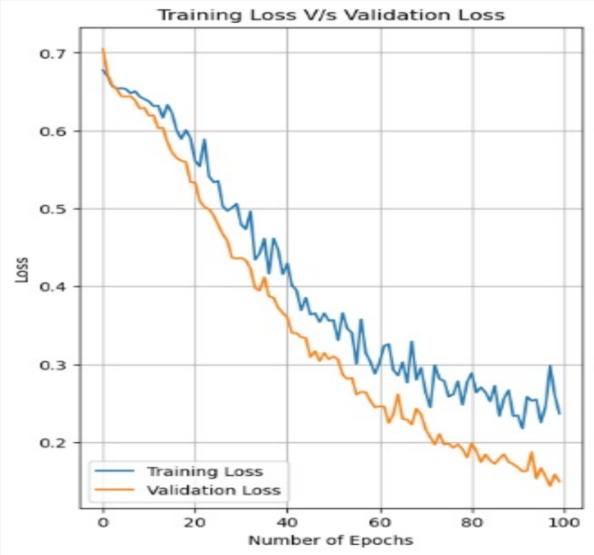
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 92, 140, 32)       896

 max_pooling2d (MaxPooling2D  (None, 46, 70, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 46, 70, 32)        9248

 max_pooling2d_1 (MaxPooling  (None, 23, 35, 32)       0
 2D)

 flatten (Flatten)           (None, 25760)             0

 dropout (Dropout)           (None, 25760)             0

 dense (Dense)               (None, 50)                1288050

 dropout_1 (Dropout)         (None, 50)                0

 dense_1 (Dense)             (None, 2)                 102

=================================================================
Total params: 1,298,296
Trainable params: 1,298,296
Non-trainable params: 0
```

Advanced CNN Model

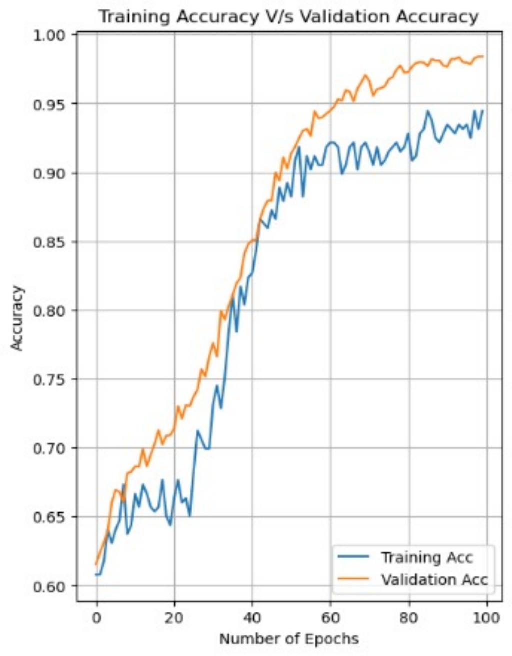
```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 92, 140, 32)       896

 max_pooling2d (MaxPooling2D  (None, 46, 70, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 46, 70, 32)        9248

 max_pooling2d_1 (MaxPooling  (None, 23, 35, 32)       0
 2D)

 conv2d_2 (Conv2D)           (None, 23, 35, 64)        18496

 max_pooling2d_2 (MaxPooling  (None, 11, 17, 64)       0
 2D)

 conv2d_3 (Conv2D)           (None, 11, 17, 128)       73856

 max_pooling2d_3 (MaxPooling  (None, 5, 8, 128)        0
 2D)

 dropout (Dropout)           (None, 5, 8, 128)         0

 flatten (Flatten)           (None, 5120)              0

 dropout_1 (Dropout)         (None, 5120)              0

 dense (Dense)               (None, 50)                256050

 dropout_2 (Dropout)         (None, 50)                0

 dense_1 (Dense)             (None, 2)                 102

=================================================================
Total params: 358,648
Trainable params: 358,648
Non-trainable params: 0
```

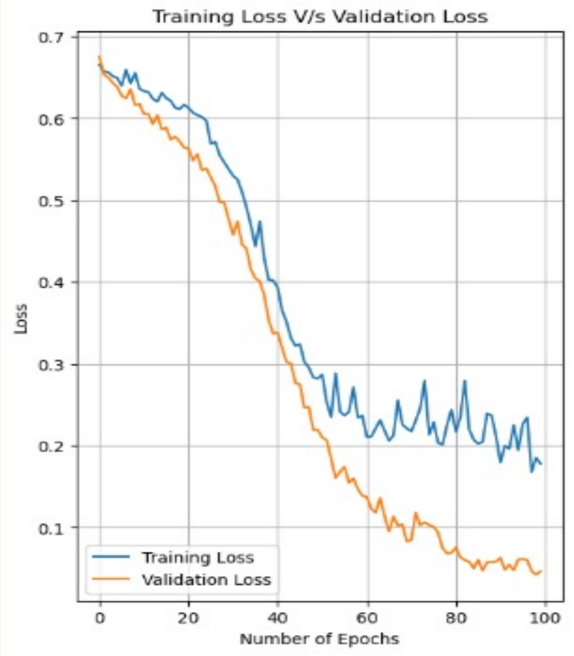
Performance Graphs (MIAS Dataset)





(Performance with Baseline CNN Model)

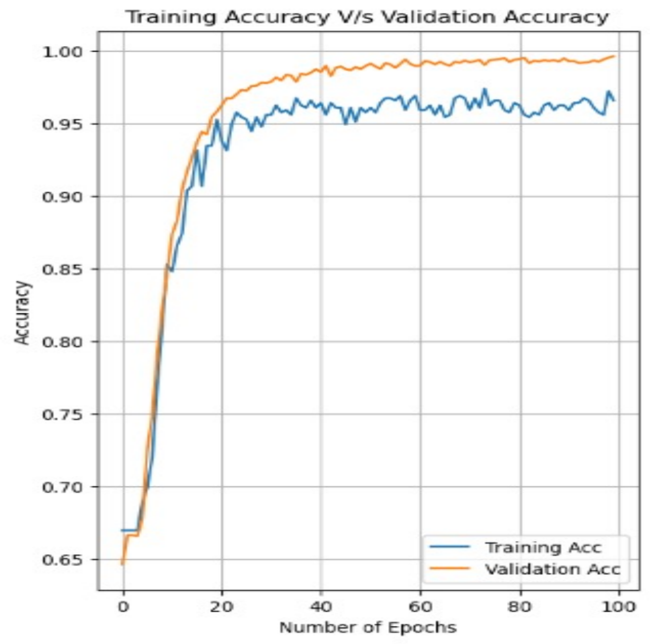


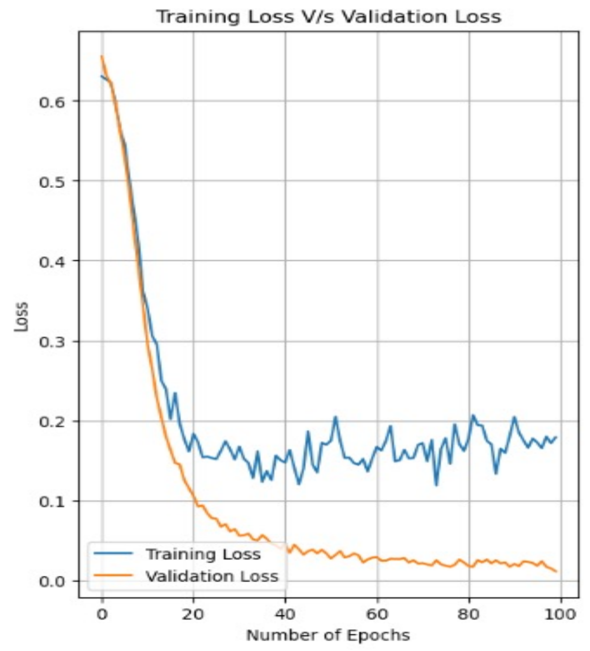


(Performance with Advanced CNN Model)

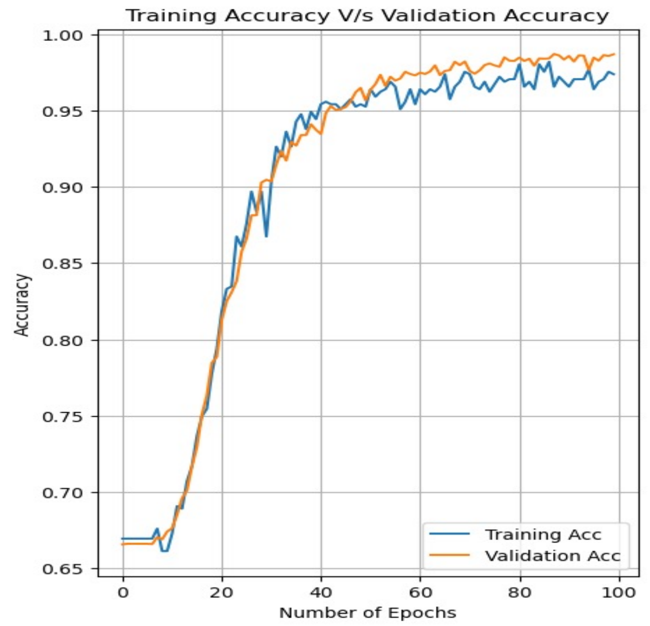
Performance Graphs (INbreast Dataset)

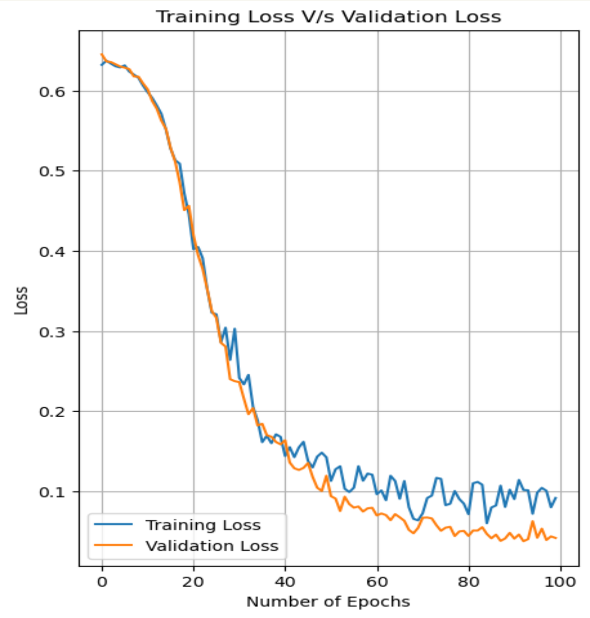
(Performance with Baseline CNN Model)





(Performance with Advance CNN Model)





Classification Report

MIAS Dataset

Baseline CNN

	precision	recall	f1-score	support
0	0.91	0.90	0.90	286
1	0.94	0.94	0.94	478
accuracy			0.93	764
macro avg	0.92	0.92	0.92	764
weighted avg	0.93	0.93	0.93	764

Advanced CNN

	precision	recall	f1-score	support
0	0.93	0.88	0.91	286
1	0.93	0.96	0.95	478
accuracy			0.93	764
macro avg	0.93	0.92	0.93	764
weighted avg	0.93	0.93	0.93	764

Classification Report

INBreast Dataset

Baseline CNN

	precision	recall	f1-score	support
0	0.97	0.98	0.98	1044
1	0.95	0.94	0.95	483
accuracy			0.97	1527
macro avg	0.96	0.96	0.96	1527
weighted avg	0.97	0.97	0.97	1527

Advanced CNN

	precision	recall	f1-score	support
0	0.95	0.99	0.97	1044
1	0.98	0.88	0.93	483
accuracy			0.96	1527
macro avg	0.96	0.93	0.95	1527
weighted avg	0.96	0.96	0.95	1527

Comparing Accuracies

	MIAS Dataset	IN Breast Dataset
Baseline CNN	91.75%	92.54%
Advanced CNN	93.14%	94.54%

Conclusion & Future scope

	A	B	C	D
1	Year	Method Used	Accuracy(%)	Error Rate
2	2017	K-Nearest Neighbor [12]	83 to 86	19.28
3	2019	Pre-Trained Networks [10]	90 to 97	4.74
4	2017	Feature Extracted Using CNN	83 to 90	4.28
5	2018	Deep Convolution Neural Network [11]	91.54	8.54

For comparison, we have compared our result (93.14% and 94.54% validation accuracy) with several published studies

1. Results are insensitive to the resolution of images.
2. Scope to implement auto-encoders instead of manually reducing image size.
3. It can compress data without losing the prominent features.

THANK YOU!!