INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT (IJSREM)
VOLUME: 08 ISSUE: 04 | APRIL - 2024 SJIF RATING: 8.448 ISSN: 2582-3930

Classification of Benign and Maligant using Deep Learning Technique

K.Vishnavi[1] M.Mahendra[2] I.sonia[3] M.Karthik[4]

[1,2,3,4]ECE, RVRJCCE, INDIA

E-mail: vyshnavikaku@gmail.com[1]
E-mail: munthamahendrasai@gmail.com[2]
E-mail: soniaimmadi123@gmail.com[3]
E-mail: macherlakarthik99 @gmail.com[4]

Abstract

Breast cancer is one of the most common malignant tumors in women, which seriously affect women's physical and mental health and even threat to life. At present, microscopic images are the important criterion for doctors to diagnose breast cancer. However, due to the complex structure of microscopic images, it is relatively difficult for doctors to identify breast cancer features. Deep learning is the most mainstream image classification algorithm which uses Convolutional Neural Network (CNN). In this project first, the BreakHis400x (Breast Cancer Histopathological Images) images will undergo pre-processing such as enhancement and data augmentation in order to increase the number of images. Secondly, the pre-processed dataset will be trained on deep learning model that extracts the features and generate the performance evaluation metric of CNN models. We are going to use python software for classifying the images.

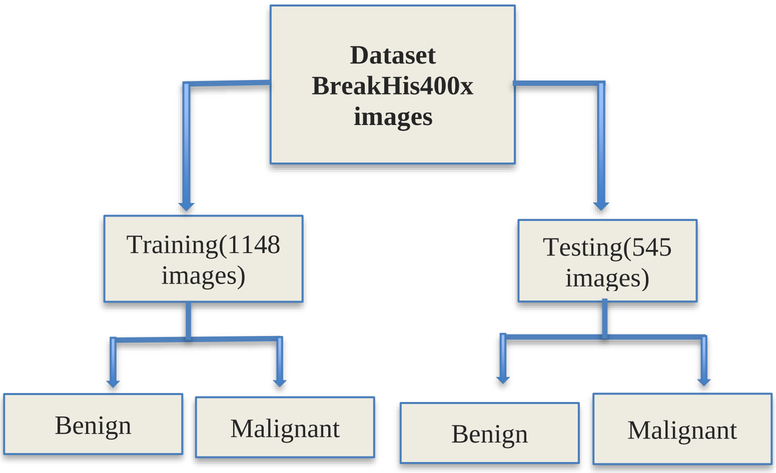
1. INTRODUCTION

Breast cancer has become increasingly prevalent among women worldwide, with its incidence on the rise in both developed and developing nations. Clinically, malignant tumors are categorized as positive, while benign tumors are considered negative. Various methods are employed for breast cancer detection, including mammography, computed tomography, photoacoustic imaging, nuclear magnetic resonance imaging, microwave imaging, and others.

Microscopic image classification in breast cancer involves the application of deep learning techniques to analyze and categorize cellular structures and tissue patterns visible under a microscope. These techniques aim to assist in the detection and diagnosis of breast cancer by automatically identifying various features indicative of malignancy or benignity in histopathological images. Deep learning models, such as convolutional neural networks (CNNs), are then trained on a dataset of labeled breast cancer images. This dataset consists of images annotated by pathologists to indicate the presence or absence of cancerous cells.

This paper conducts a detailed comparison of various deep learning models with the goal of enhancing their accuracy in diagnosing breast cancer. Additionally, the study meticulously analyzes confusion matrices generated for desired model, providing insights into their classification outcomes, including true positives, true negatives, false positives, and false negatives Roc, F-score, Recall. Through this evaluation the paper aims to identify the most effective deep learning architecture and classifies the given image into benign and malignant.

2. DATASET:



Some of the sample data set images of Breakhis400X dataset are as follows:

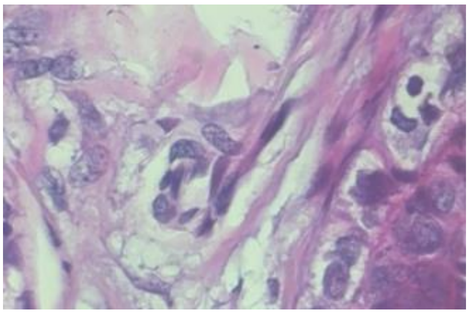


Fig1.(a)

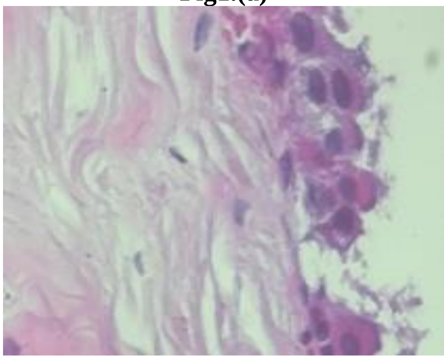


Fig1.(b)

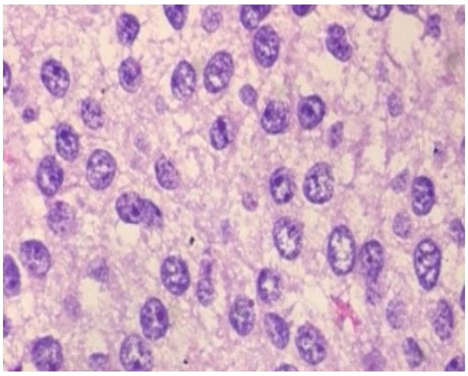


Fig1.(c)

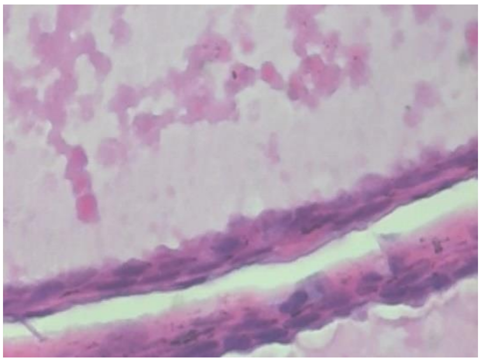


Fig1.(d)

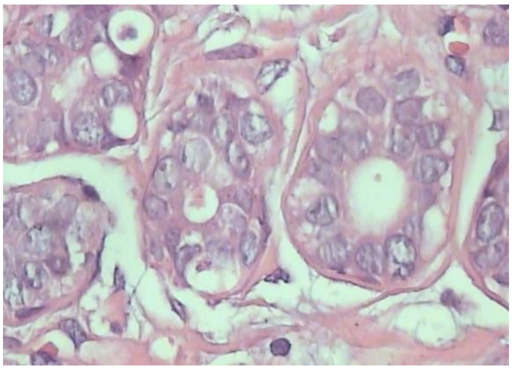


Fig1.(e)

III.METHODOLOGY

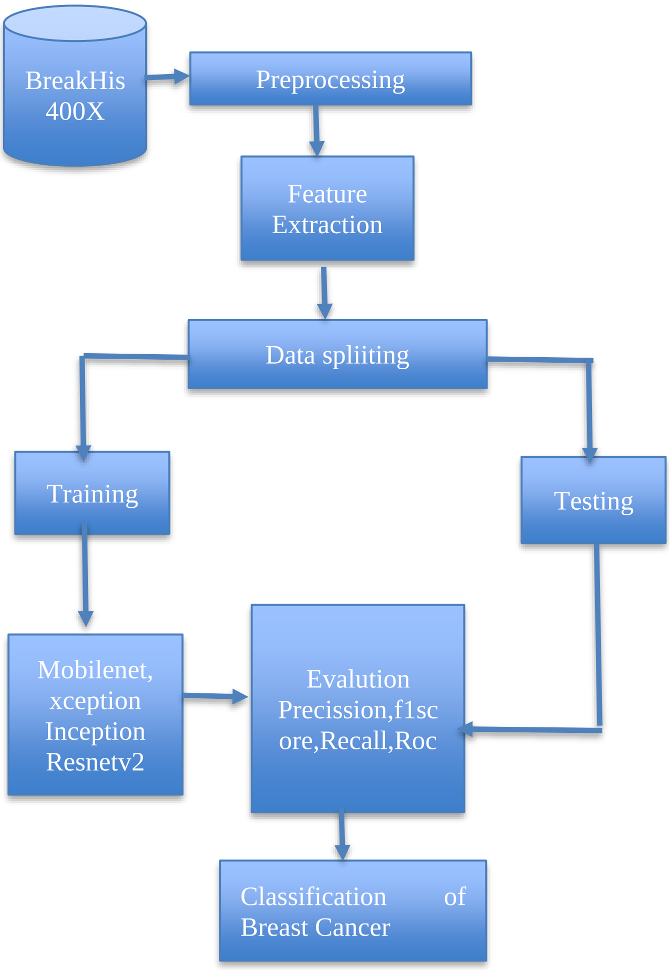


Fig-2: *Process flow diagram*

3.1 Data Preprocessing:

Pre-processing involves manipulating images at their most fundamental levels, aiming to enhance data quality by mitigating distortions or accentuating specific visual attributes. Data Augmentation, on the other hand, is a method to generate additional images, utilizing techniques like random horizontal flipping, resize cropping, and rotation. The interpretation of final data processing outcomes can be significantly influenced by the approach to image pre-processing.

3.2 Fine Tuning

In this study, fine-tuning was employed to enhance the performance of large pre-trained convolutional models specifically for the BreakHis 400x image dataset. Fine-tuning involves adapting a pre-trained model, initially trained on a comprehensive dataset like ImageNet, to the unique characteristics of the BreakHis dataset. This process allows the model to extract new features relevant to classifying breast histopathology images while leveraging the knowledge acquired from the broader dataset[3]. The fine-tuning process consists of two stages: pre-training on ImageNet to learn a versatile set of features and then further training on the BreakHis dataset to refine these features for the specific task at hand. Careful consideration of hyperparameters, such as the number of layers to be frozen, learning rate, and batch size, is crucial to strike a balance between adapting the model to the new dataset and preventing overfitting. By fine-tuning on the BreakHis 400x images, the model can efficiently learn to classify breast histopathology images with improved accuracy and effectiveness.

3.3 Image Augementation:

Image augmentation is a pivotal strategy in deep learning, particularly beneficial when dealing with limited training data. It involves applying various transformations to existing images to expand the dataset artificially. Common techniques include rotation, flipping, zooming, cropping, translation, brightness and contrast adjustment, noise injection, and color jitter. By incorporating these transformations, the effective size of the training dataset increases, promoting better generalization and robustness of deep learning models. This augmentation aids in capturing diverse variations of the input data, thereby enhancing the model's ability to recognize patterns and features across different conditions. Moreover, augmented data can help mitigate overfitting by introducing variability into the training process. The manually labeled training data set is limited. Due to the limitation of the training samples, CNN is prone to over-fitting during the training process, resulting in low medical image recognition rate and unsatisfactory diagnosis results. Therefore, this requires a large increase in the training data set. The solution is to enhance the images dataset by affine transformation and random cropping

3.4Feature Extraction:

Extract features from the intermediate layers of the pretrained model. These layers capture hierarchical features of increasing complexity. Commonly used layers for feature extraction are the convolutional layers before the fully connected layers

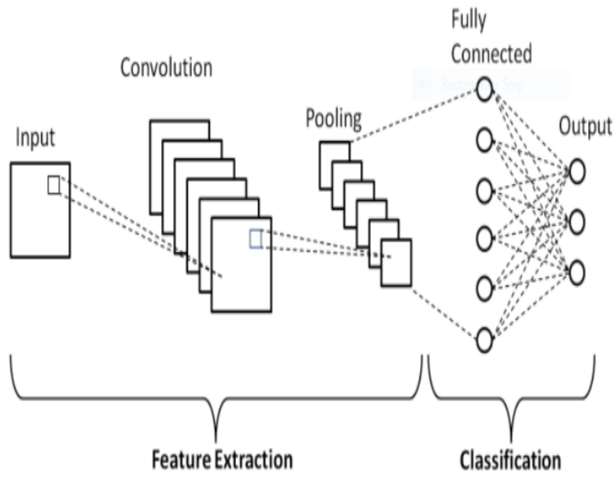


Fig-3: *CNN Architecture*

A Convolutional Neural Network (CNN) architecture comprises several layers designed to process input images: an input layer representing raw pixel data, followed by convolutional layers that use filters to extract features like edges or textures, with each filter applied across the input image. Activation functions like ReLU introduce non-linearity after convolutions. Pooling layers down sample feature maps to reduce spatial dimensions by using operations like max or average pooling. Fully connected layers process high-level features for classification, with each neuron connected to all neurons in the preceding layer. Before passing to fully connected layers, feature maps are flattened into one-dimensional vectors. The output layer produces final predictions or scores, with the number of neurons depending on the task, such as binary or multi-class classification. Popular CNN architectures, like MobileNet, AlexNet, VGG, GoogLeNet, ResNet, Exception and DenseNet, vary in depth, width, and layer configurations based on specific tasks, dataset size, and computational resources available.

MobileNet:

MobileNet emerges as a versatile and efficient convolutional neural network (CNN) architecture, well-suited for real-world applications. It distinguishes itself by leveraging depthwise separable convolutions, a novel technique that replaces traditional convolutions, resulting in lighter models with reduced computational demands. Notably, MobileNets introduce two innovative global hyperparameters: the width multiplier and resolution multiplier. These parameters offer developers flexibility to optimize models based on their specific requirements, balancing between speed, accuracy, and model size. Built upon depthwise separable convolution layers, MobileNets consist of depthwise convolutions followed by pointwise convolutions, totaling 28 layers when counted separately. Despite their depth, MobileNets maintain efficiency, with a standard configuration comprising approximately 4.2 million parameters[2]. Moreover, by adjusting the width multiplier, developers can further reduce model complexity, making MobileNet highly adaptable for resource-constrained environments without compromising performance.

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5× Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
5× Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Table-1: *MobileNet Body Architecture*

Depthwise seperable convolutional layer:

A single Depthwise separable convolution unit looks like this:

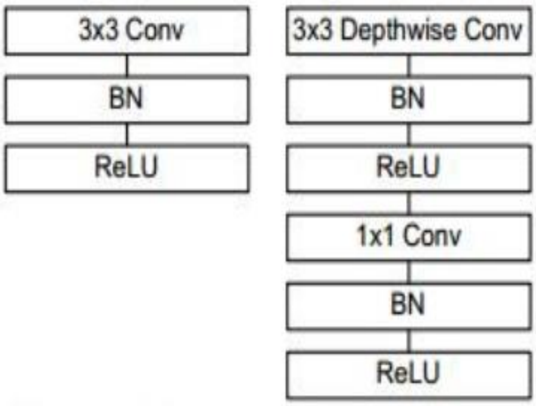


Fig-4: *Single Depth-wise separable convolution*

1. Depth- wise convolution is the channel-wise DK×DK spatial convolution. Suppose in the figure above, we have 5 channels, then we will have 5 DK×DK spatial convolution.

2.Pointwise convolution actually is the 1×1 convolution to change the dimension. With the above operation, the operation cost is:

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F$$

where M is: Number of input channels, N is: Number of output channels, DK: Kernel size and DF is: Feature map size.

For standard convolution, it is:

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F$$

Thus, the computation reduction is:

$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F}$$
$$= \frac{1}{N} + \frac{1}{D_K^2}$$

When DK×DK is 3×3, 8 to 9 times less computation can be achieved, but with an only a small reduction in accuracy. Standard Convolution vs Depthwise Separable Convolution for ImageNet dataset:

Width Multiplier:

The width multiplier (denoted by α) is a global hyperparameter that is used to construct smaller and less computationally expensive models. Its value lies between 0 and 1. For a given layer and value of α, the number of input channels 'M' becomes α * M and the number of output channels 'N' becomes α * N hence reducing the cost of computation and size of the model at the cost of performance. The computation cost and the number of parameters decrease roughly by a factor of α2. Some commonly used values of α are 1,0.75,0.5,0.25.

$$D_K \cdot D_K \cdot \alpha M \cdot D_F \cdot D_F + \alpha M \cdot \alpha N \cdot D_F \cdot D_F$$

Resolution Multiplier:

The second parameter introduced in MobileNets is called the resolution multiplier and is denoted by ρ. This hyperparameter is used to decrease the resolution of the input image and this subsequently reduces the input to every layer by the same factor. For a given value of ρ, the resolution of the input image becomes 224 * ρ. This reduces the computational cost by a factor of ρ2.

$$D_K \cdot D_K \cdot \alpha M \cdot \rho D_F \cdot \rho D_F + \alpha M \cdot \alpha N \cdot \rho D_F \cdot$$

Xception:

The Xception architecture, an abbreviation for "Extreme Inception," is a sophisticated neural network renowned for its innovative design and exceptional performance in various image-related tasks. Comprising 36 convolutional layers, Xception forms a robust feature extraction backbone crucial for image classification tasks. In our experimental evaluation, we focus exclusively on image classification, where the convolutional base is followed by a logistic regression layer. Optionally, fully-connected layers can be inserted before the logistic regression layer, offering flexibility in model architecture. The 36 convolutional layers are organized into 14 modules, each incorporating linear residual connections, except for the first and last modules. Essentially, Xception adopts a linear stack structure of depthwise separable convolution layers with residual connections, rendering the architecture highly adaptable and straightforward to customize for specific applications or tasks. This unique combination of depthwise separable convolutions and residual connections contributes to Xception's effectiveness and ease of modification, making it a compelling choice for a wide range of image processing tasks.

The data first goes through the entry flow, then through the middle flow which is repeated eight times, and finally through the exit flow. Note that all Convolution and SeparableConvolution layers are followed by batch normalization

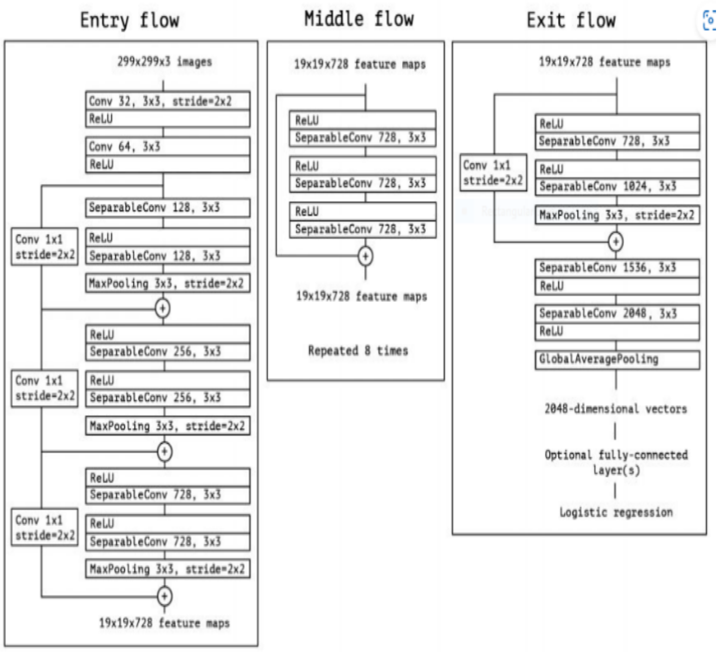


Fig-5: *overall architecture*

Depthwise seperable Convolution:

Depthwise Separable Convolutions are alternatives to classical convolutions that are supposed to be much more efficient in terms of computation time.

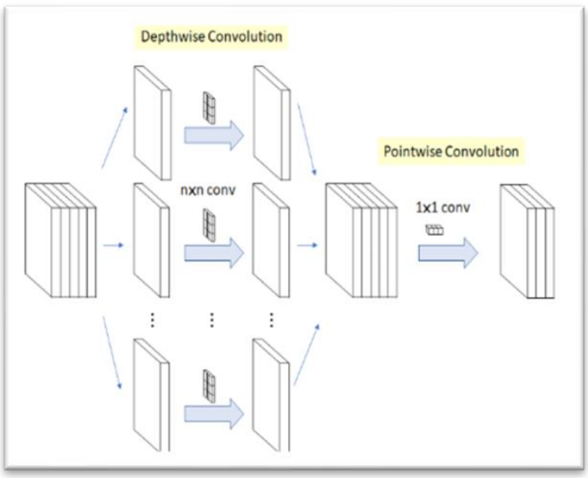


Fig-6: *Depthwise seperable convolution*

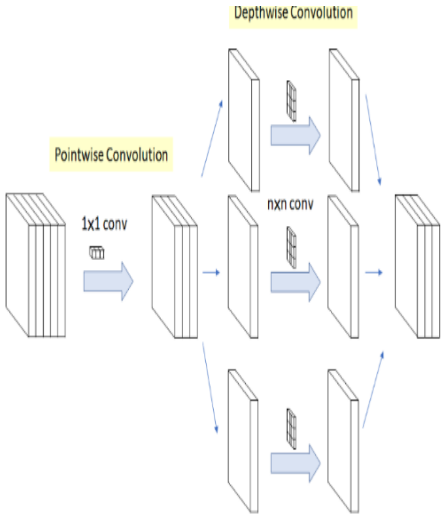


Fig-7: *Modified Depthwise seperable convolution*

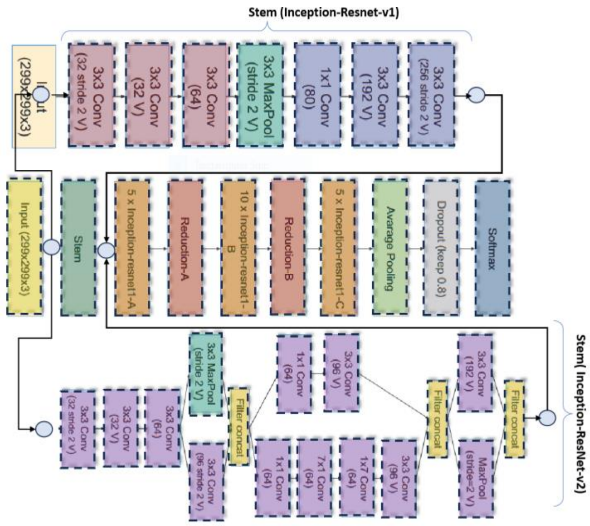
The modified depthwise separable convolution involves rearranging the order of operations compared to the original. Instead of starting directly with the depthwise convolution followed by the pointwise convolution, as in the conventional depthwise separable convolution, it begins with a pointwise convolution, followed by the depthwise convolution.

This alteration draws inspiration from the design philosophy behind the inception module in Inception-v3. In Inception-v3, a 1×1 convolution precedes any n×n spatial convolutions (typically n=3). This initial 1×1 convolution helps in reducing the dimensionality of the input feature maps before applying more computationally intensive convolutions.

By employing this modified approach, the network benefits from dimensionality reduction early on, which can enhance computational efficiency and aid in capturing more complex features effectively.

InceptionResnetV2:

The InceptionResNetV2 architecture merges the best features of the Inception and ResNet models to create a hybrid network that excels in image recognition tasks, particularly in the medical domain. By combining the strengths of both models, such as faster training times and the ability to mitigate the vanishing gradient problem, InceptionResNetV2 achieves superior performance. Its incorporation of residual connections enables the network to skip certain layers during training, facilitating the flow of gradients and allowing for the creation of deeper architectures. Additionally, the utilization of multiple-size kernels within a single layer enhances the model's capacity to capture features of varying complexity across different scales and hierarchies. With its distinctive design comprising convolutional layers, filter concatenation, ReLU activation functions, ResNet structures, and inception modules, InceptionResNetV2 has demonstrated remarkable efficacy in medical image classification tasks. This innovative approach has positioned the model as a foundation for cutting-edge research and practical applications in the field of computer vision.



RESULTS AND DISCUSSIONS:

The following figure is a confusion matrix:

A confusion matrix provides a snapshot of a classification model's performance by comparing predicted and actual class labels in a tabular format. It condenses results into true positives (correct positive predictions), true negatives (correct negative predictions), false positives (incorrect positive predictions), and false negatives (incorrect negative predictions). In binary classification, it typically comprises a 2x2 table, with rows representing predicted labels and columns representing actual labels. This concise summary allows for a quick assessment of the model's accuracy and errors, aiding in decision-making for model refinement or deployment.

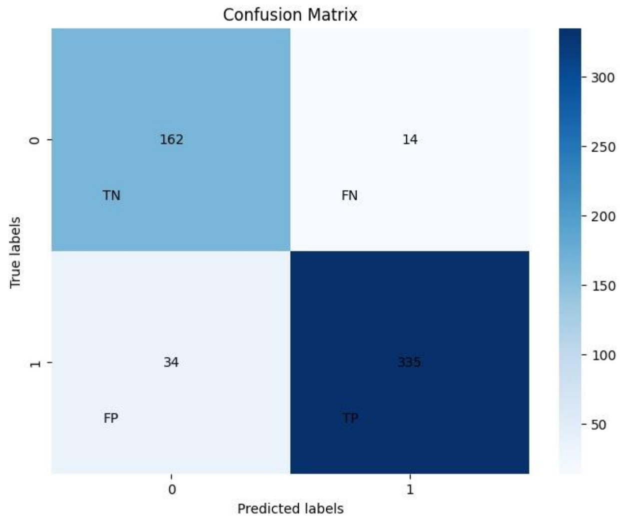


Fig-8: *confusion matrix of xception model*

Performance Metrics:

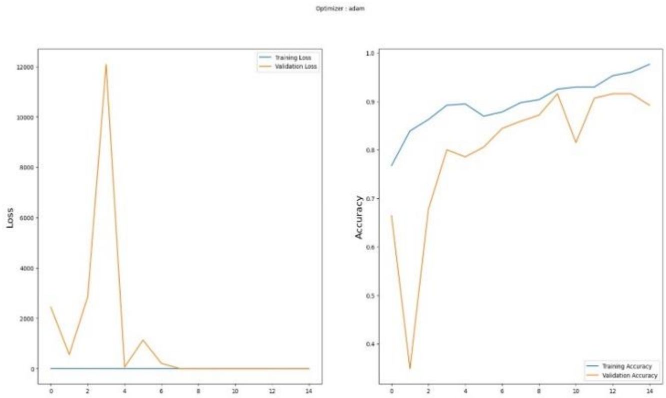
Metric	Formula	Xception
Accuracy	TP+TN / TP+TN+FP+FN	91.55
Precision	TP / TP+FP	98.64
Recall	TP / TP+FN	89.87
ROC AUC score	using TPR, FPR	93.15
F1-Score	2*precision*recall / Precision+recall	94.05

Table-2: *Results of Xception Model:*

Method	Accuracy	Error rate
MobilNet	88.44	0.726
InceptionV2	89.17	0.298
Xception	91.55	0.391

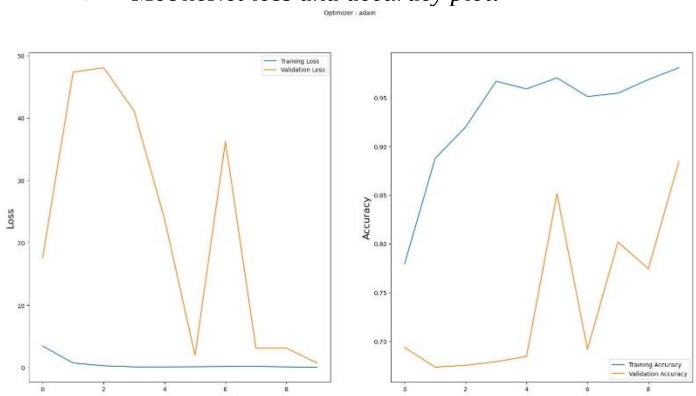
Table-3: *Comparison of Accuracy and Error rate for different models*

- *InceptionResnetV2 loss and accuracy:*



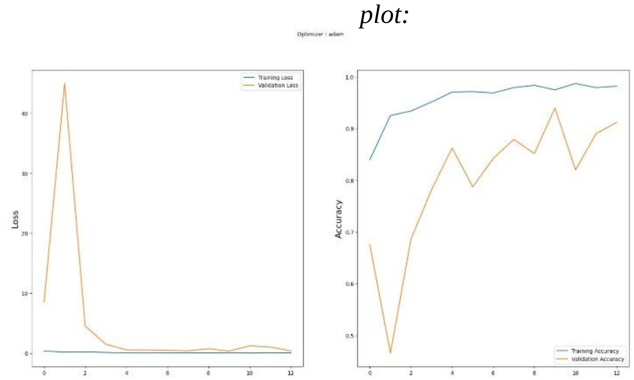
x-axis: number of epochs y-axis: loss , accuracy.

- *MobileNet loss and accuracy plot:*

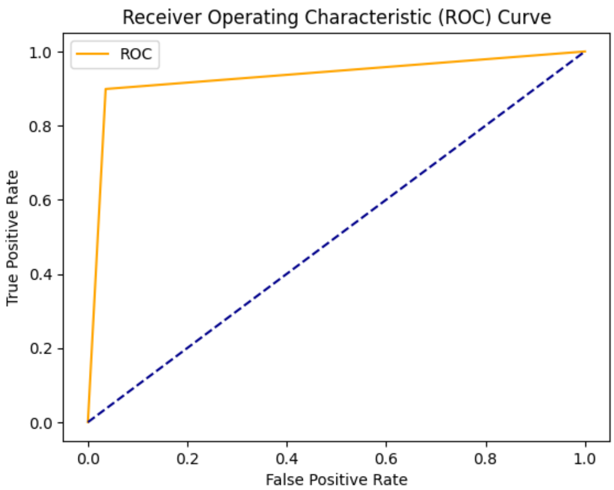


x-axis: number of epochs y-axis: loss , accuracy.

- *Xception loss and accuracy and ROC curve plot:*



x-axis: number of epochs y-axis: loss , accuracy.



CONCLUSION:

In this paper, we proposed the comparative study of xception deep learning model's performance in the classification of breast cancer tumor. We performed binary classification using BreakHis400x dataset. The suggested model i.e. Xception extensively compared to other two CNN models MobilNet and InceptionResNetV2. Xception shows highest accuracy compared to other two models. The proposed approach shows 91.55% accuracy only by using 400x magnification images in the dataset. Furthermore, we investigated the impact of transfer learning in improving the performance of deep convolutional models.

REFERENCES:

[1] Xception: Deep Learning with Depthwise Separable Convolutions Francois Chollet

[2] MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications Andrew G. Howard Menglong Zhu Bo Chen Dmitry Kalenichenko Weijun Wang Tobias Weyand Marco Andreetto Hartwig Adam2022, MIMSE 2022 (Informatics and Computer Science), pp.398, 2022.

[3] Hybrid Inception Architecture with Residual Connection: Fine-tuned Inception-ResNet Deep Learning Model for Lung Inflammation Diagnosis from Chest Radiographs MehdiNeshata,, Muktar Ahmedb,c, Hossein Askarid , Menasha Thilakaratnee , and Seyedali Mirjalili

[4] SHARAF J.MALEBARY,IEEE Access Automated Breast cancer classification system using Deep learning and Ensembles Learning in Digital Mammogram,vol.No,2021

[5] Tariq Mahmood ,Breast lessions classifications of mammographic images using a convolutional neural network-based approach,vol.No,2022

[6] Kiran Jabeen,Breast Cancer Classification from Mammogram Images Using Enhanced Deep Learning,vol.No,2023

[7] Md Zahangir Alom, Chris Yakopcic, Tarek M. Taha, and Vijayan K. Asari
Department of Electrical and Computer Engineering, University of Dayton, OH, USA