

Classification of Histopathological Breast cancer images using Deep Learning

B.Tech Project Report submitted

in partial fulfilment of the requirements for the degree of

Bachelor of Technology

in

Electrical Engineering

by

Rishab Kumar

1701EE39

under the supervision of

Dr. M H Kolekar



DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY PATNA

May, 2021

Certificate

This is to certify that the dissertation entitled “Classification of Histopathological Breast cancer images using Deep Learning” submitted by Rishab Kumar (Roll No. 1701EE39) to Indian Institute of Technology Patna, in partial fulfilment of the requirement for the award of the degree of Bachelor of Technology in Electrical Engineering is a bona-fide work carried out under my supervision. The dissertation fulfils the requirements as per the regulations of this Institute and in my opinion meets the necessary standards for submission. The contents of this dissertation have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Dr. M H Kolekar
Associate Professor
Department of Electrical Engineering
Indian Institute of Technology Patna

Declaration

This is to certify that

1. The work contained in this report has been done by me under the guidance of my supervisor.
2. The work has not been submitted to any other Institute for any degree or diploma.
3. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
4. My project supervisor has reviewed this version of the thesis. After necessary revisions have been done, the document has been approved for submission before the department.

Date: May, 2021

(Rishab Kumar)

Place: Bihta, Patna

(1701EE39)

Abstract

Breast Cancer (BC) is quite common in present times and it is responsible for many deaths among women all over the world. As a result, early and accurate detection of breast cancer has become critical for accurate diagnosis. By studying tissue slides, specialized clinicians can identify malignant tissues, but the accuracy is not acceptable in the medical industry. As a result, artificial models are required to identify the images with greater precision. Several studies have created automated methods and methodologies to predict the presence of breast cancer utilizing various medical imaging modalities. CNNs (Convolutional Neural Networks) are deep learning models that have seen a lot of success in the field of medical imaging and can be used to detect breast cancer. In this paper, I present a method for classifying breast tissue images into malignant and non-cancerous using a hybrid convolutional neural network.

Key words: CNN, Breast Cancer, Deep Learning, Histopathology, Binary Classification

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor Dr. Mahesh H Kolekar for providing his constant support, invaluable guidance and inspiring attitude throughout the project. I wish to express my gratitude to my colleagues, for helpful discussions and introducing me to this vast research field. I am also thankful to all my friends and family for their direct and indirect support in this project.

List of Figures

1.1	Examples of Histopathological images for breast cancer	13
1.2	Examples of Mammogram images for breast cancer	14
1.3	Examples of MRI images for breast cancer	14
1.4	Flowchart showing classes and subclasses for breast images	15
2.1	Images from BreakHis dataset for 40x magnification	17
2.2	Images from ICIAR2018 challenge dataset	18
2.3	Images from Kaggle Breast Histology dataset	19
3.1	Structure of DenseNet	21
3.2	Structure of ResNets	22
3.3	Structure of VGG16	24
3.4	Overall Architecture of Xception network	25
4.1	Flowchart for the proposed model	28
4.2	Architecture of the proposed model plotted by code	29
4.3	Max Pooling	31
4.4	Information about the layers of the model	35
5.1	Accuracy and validation accuracy for 100 epochs training of the model	37
5.2	Loss and validation loss for 100 epochs training of the model	37
5.3	Confusion Matrix for different magnification factors for Breakhis dataset	41
5.4	Confusion Matrix for a) ICIAR2018 dataset b) Kaggle BHI dataset	41

List of Tables

2.1	Structure of the BreakHis Dataset	17
3.1	Information about different versions of DenseNet	20
3.2	Information about different versions of DenseNet	22
3.3	Information about different versions of VGG	23
3.4	Information about layers of Inception_v3	26
5.1	Information about the number of training and testing images for Breakhis, ICIAR2018 and Kaggle BHI dataset	37
5.2	Information about the performance values (accuracy, sensitivity (recall), specificity, precision and F1-score for different datasets as obtained from the proposed model	40
6.1	Comparison of the proposed model with other existing papers	42

Contents

Certificate	3
Declaration	4
Abstract	5
Acknowledgements	6
List of Figures	7
List of Tables	8
 1. Introduction	12
1.1. Overview	12
1.2. Breast Cancer	12
1.3. Types of Images	13
1.3.1. Histopathological Images	13
1.3.2. Mammogram Images	13
1.3.3. MRI images	14
1.3.4. Classes of Images	15
1.4. Magnification Factors	15

2.	Datasets	16
2.1.	Breakhis Dataset	16
2.2.	ICIAR2018 Dataset	17
2.3.	Kaggle Breast Histology Images	19
3.	CNN Modules	20
3.1.	DenseNets	20
3.2.	ResNets	22
3.3.	VGG	24
3.4.	Xception	25
3.5.	Inception	26
4.	Methodology	27
4.1.	Overview	27
4.2.	Abstract	27
4.3.	Model Architecture	28
4.3.1.	Input Details.	30
4.3.2.	Convolution Layers.	30
4.3.3.	Pooling.	30
4.3.4.	Fully connected Layer.	31
4.3.5.	Dense Layer.	32
4.3.6.	Dropout Layer.	32
4.3.7.	Global Average Pooling Layer	33

4.3.8.	Batch Normalisation Layer	33
4.3.9.	Concatenation Layer	33
4.4.	Layers in the Model	34
5.	Implementation and Results	36
5.1.	Training.	36
5.2.	Testing.	37
5.3.	Experimental Results	38
5.3.1.	Performance Quantities	38
5.3.2.	Accuracy of the Trained Network.	40
5.3.3.	Confusion Matrix.	41
6.	Conclusions	43
References		45
Web References		48

Chapter 1

Introduction

1.1 Overview

In this chapter we are going to discuss about the breast cancer images which are used in medical science. We will discuss the types of images available, classes of the images which are to be categorised, and we will also talk about different standard magnification factors of the images.

1.2 Breast Cancer

Breast cancer is a condition that results from the existence of malignant tissue in a breast tumor. According to the World Health Organization's (WHO) International Agency for Research on Cancer (IARC), cancer claimed the lives of 8.2 million people in 2012. According to the same data, there would be 27 million new instances of this disease until 2030. Because there is a potential of developing breast cancer if a tumor is present in the breast tissue, early detection and diagnosis of the cancer has become critical.

1.3 Types of Images

There are many types of breast tissue available as a standard dataset. Among these, the most used are listed below.

1.3.1 Histopathological Images

Pathology digitization is a popular trend that allows enormous volumes of visual data to be analyzed automatically. The images taken directly from the microscope with various magnifications are known as histopathology pictures. The photos are stained with hematoxylin and eosin, which gives them a pink color.

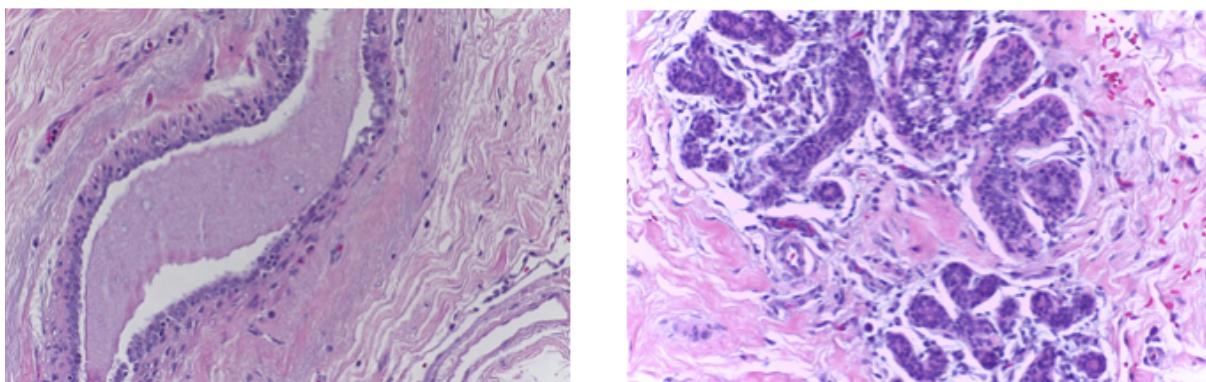


Figure 1.1: Examples of Histopathological images for breast cancer

1.3.2 Mammogram Images

Mammograms are X-ray images of the breasts that can be created using either film-screen or digital mammography. The woman getting a mammogram must position her breast between the two plates, which will compress it together to keep it in place and generate a precise image. Squeezing the breast flattens it for a better image and prevents blurring.

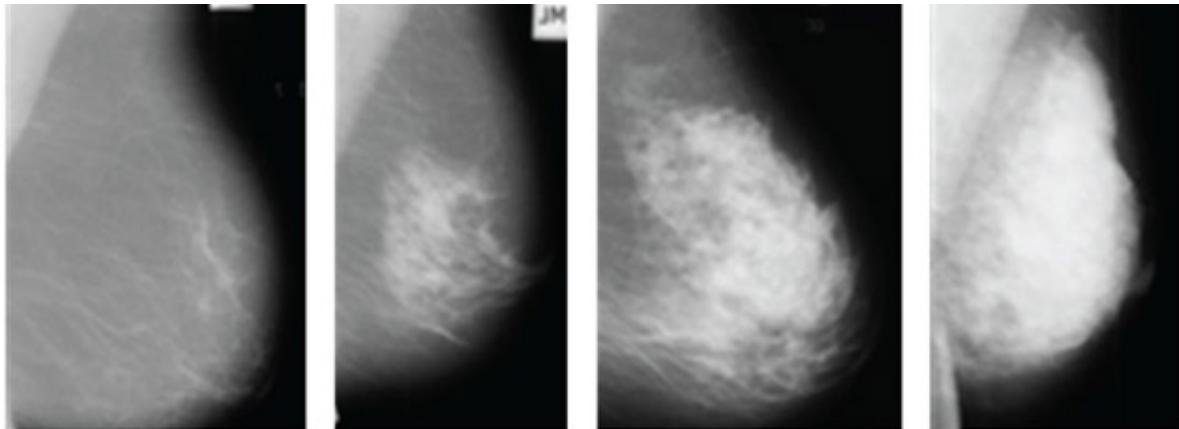


Figure 1.2: Examples of Mammogram images for breast cancer
<https://www.healthline.com/health/breast-cancer/mammogram-images-breast-cancer>

1.3.3 MRI Images

MRI can also be used to obtain images of breast tissue (Magnetic Resonance Imaging). According to new research, MRI can detect tiny breast lesions that are sometimes missed by mammography. It can also assist people with breast implants detect breast cancer. It can also be used to treat younger women with thick breast tissue. It has a number of drawbacks, including the inability to discern between different forms of breast cancer abnormalities.

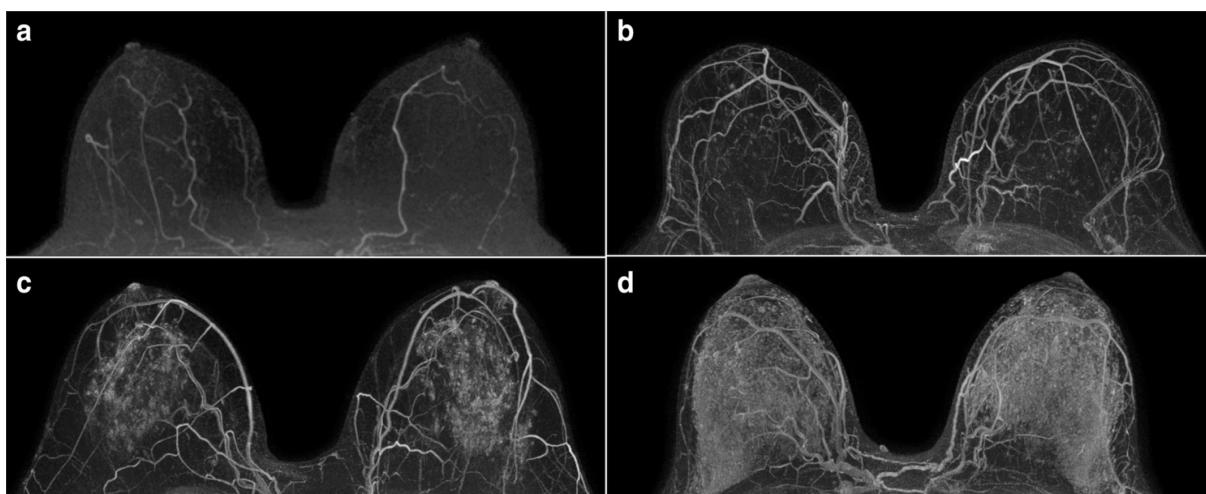


Figure 1.3: Examples of MRI images for breast cancer
<https://onlinelibrary.wiley.com/doi/abs/10.1002/jmri.26762>

1.4 Types of Images

The primary distribution of a tumour tissue of a breast image is benign (non-cancerous) and malignant (cancerous). If the tissue does not contain a tumour then it is called normal tissue. The malignant tissue is further subdivided into 2 types of cancer named as in-situ (Ductal carcinoma) and invasive.

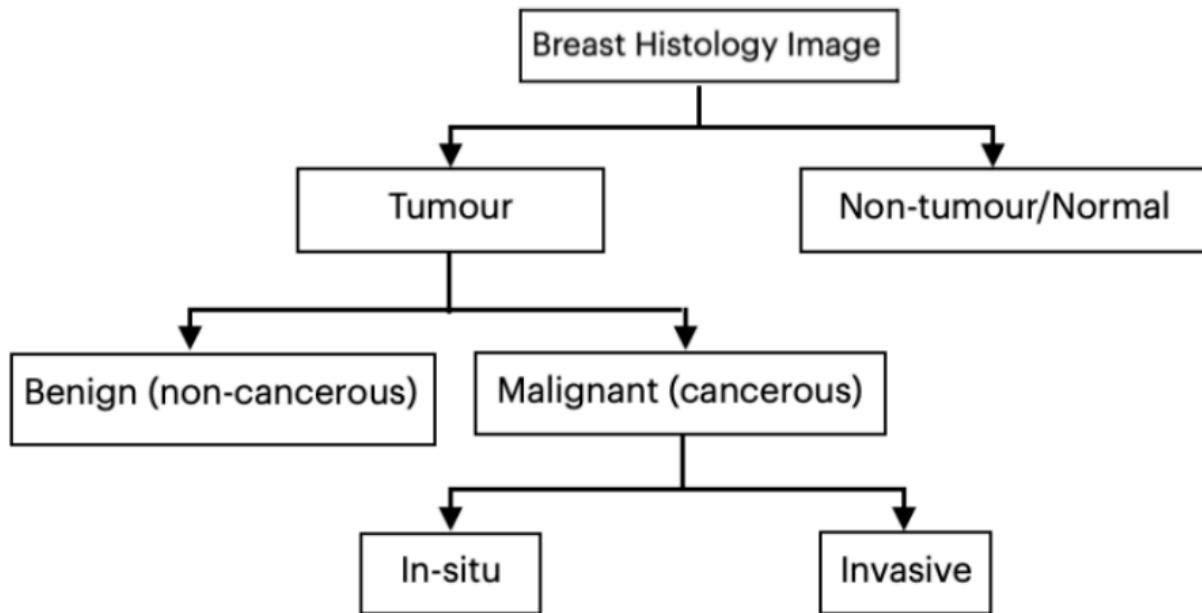


Figure 1.4: Flowchart showing classes and subclasses for breast images

1.5 Magnification Factors

The histopathological images have 4 standard magnification sizes. These are 40x, 100x, 200x and 400x. This means the actual tissue is magnified to the given number of times. The accuracies are calculated separately for all the 4 magnification sizes.

Chapter 2

Datasets

There are many available datasets for breast histology images. Most of the datasets are available as open source and can be downloaded for any type of projects/research. The most common dataset is BreakHis[9] dataset. Other datasets used in this work are ICIAR2018 dataset and a dataset taken from Kaggle.

2.1 Breakhis Dataset

This dataset [9] (available at <https://web.inf.ufpr.br/vri/databases/%20breast-cancer-histopathological-database-breakhis/>) is accessible at <https://web.inf.ufpr.br/vri/databases/percent20breast-cancer-histopathological-database-breakhis/>). There are 9,109 microscopic photos of breast tumor tissue in this collection. It was gathered from 82 patients and contains 40x, 100x, 200x, and 400x magnifying factors. There are 2,480 benign and 5,429 malignant samples in this collection. The images are 3-channel RGB with an 8-bit depth in each channel. Each image is 700 x 460 pixels in size. The four types of benign tumors include adenosis, tubular adenoma, phyllodes tumor, and fibroadenoma, and these images are classified into four subclasses. Mucinous carcinoma, lobular carcinoma, ductal carcinoma, and papillary carcinoma are the four types of malignancy. This database was created in partnership with Parana, Brazil's P&D Laboratory – Pathological Anatomy and Cytopathology.

Classes	Subtypes	Magnification Factors				Total
		40x	100x	200x	400x	
Benign (B)	Adenosis (A)	114	113	111	106	444
	Fibroadenoma (F)	253	260	264	237	1,014
	Tubular Adenoma (TA)	109	121	108	115	453
	Phyllodes Tumor (PT)	149	150	140	130	569
Malignant (M)	Ductal Carcinoma (DC)	864	903	896	788	3,451
	Lobular Carcinoma (LC)	156	170	163	137	626
	Mucinous Carcinoma (MC)	205	222	196	169	792
	Papillary Carcinoma (PC)	145	142	135	138	560
Total		1,995	2,081	2,013	1,820	7,909

Table 2.1: Structure of the BreakHis Dataset
<https://doi.org/10.1371/journal.pone.0214587.t004>

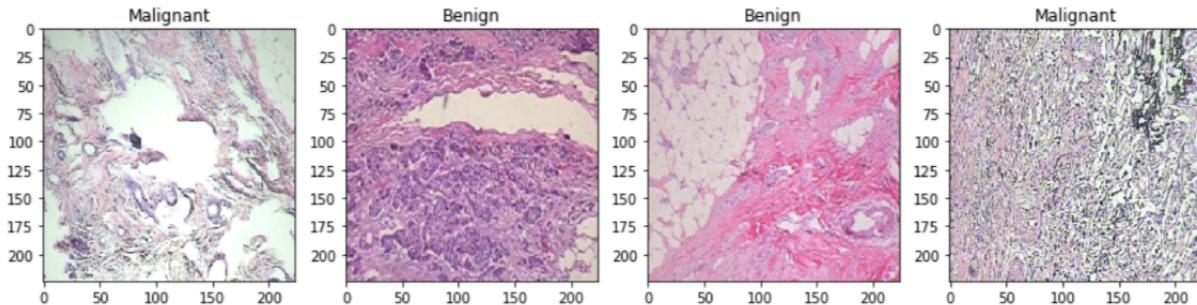


Figure 2.1: Images from BreakHis dataset for 40x magnification

2.2 ICIAR2018 Dataset

This dataset [11] (available at <https://ic iar2018-challenge.grand-challenge.org/Dataset/>) is an expanded version of the 2015 Bioimaging breast histology classification challenge dataset. The photos in the collection have a magnification factor of 200x and pixel dimensions of 0.420 m x 0.420 m. Each image is divided into four categories: (1) normal, (2) benign, (3) in situ, and (4) invasive cancer. The whole-slide photographs were annotated by two experts, and photographs that were not authorized by the experts were discarded.

Each image comprises a list of marked coordinates that encloses zones of benign, in situ carcinoma, and aggressive cancer. The purpose of this challenge was to automatically classify each image that was submitted.

The dataset contains a total of 400 microscopy images, distributed as follows:

- Normal: 100 images
- Benign: 100 images
- in situ: 100 images
- Invasive: 100 images

Microscopy images are in .tiff format and have the following specifications:

- Color model: RGB (Red, Green, Blue)
- Size: 2048 x 1536 pixels
- Pixel scale: 0.42 μm x 0.42 μm
- Memory space: 10-20 MB (approx.)
- Type of label: image-wise

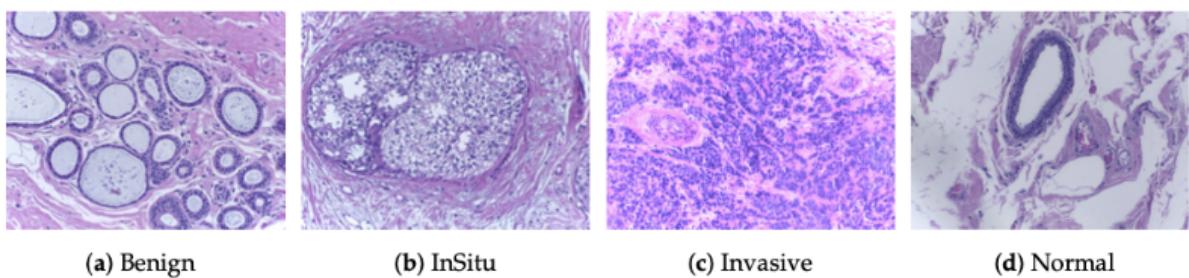


Figure 2.2: Images from ICIAR2018 challenge dataset

2.3 Kaggle Breast Histology Dataset

The original dataset [10] was made up of 162 full slide photographs of BC specimens scanned at a magnification factor of 40x. There are 277,524 photos in this dataset, with 198,738 benign and 78,786 malignant pictures. The original dataset can be found at <https://www.kaggle.com/paultimothymooney/breast-histopathology-images>.

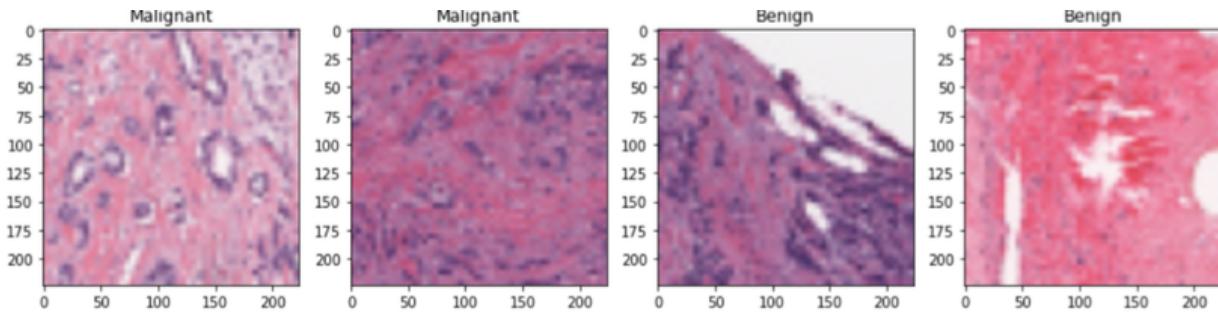


Figure 2.3: Images from Kaggle Breast Histology dataset

Chapter 3

CNN Models

This work proposes a hybrid convolutional neural network model. There exist some pre-built CNN models that have shown good results on the breast histology images. These are DenseNets, Resnets, Xception networks, Inception networks and VGG networks. 5 models were tested for the same datasets. These are DenseNets, ResNets, VGG, Xception and Inception networks. These are the best performing CNN models available in the field of medical imaging.

3.1 DenseNets

The Dense Convolutional Network (DenseNet) is a feed-forward network in which each layer is connected to the next. Each layer's inputs are the feature-maps from the previous levels. Every layer's feature-maps are used as inputs into the succeeding layers. DenseNets have a number of appealing features, including the ability to overcome the vanishing-gradient problem, encourage feature reuse, improve feature propagation, and drastically reduce the number of parameters. Our DenseNet architecture was tested on four extremely competitive object identification benchmark tests by researchers (CIFAR-10, CIFAR-100, ImageNet and SVHN). Densenets come in a variety of flavors, including DenseNet121, DenseNet169, DenseNet201, and DenseNet264. The number represents the number of layers in the network in question. The table 4.1 summarizes the differences between DenseNets versions.

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112		7 × 7 conv, stride 2		
Pooling	56 × 56		3 × 3 max pool, stride 2		
Dense Block (1)	56 × 56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56 × 56		1 × 1 conv		
	28 × 28		2 × 2 average pool, stride 2		
Dense Block (2)	28 × 28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28 × 28		1 × 1 conv		
	14 × 14		2 × 2 average pool, stride 2		
Dense Block (3)	14 × 14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14 × 14		1 × 1 conv		
	7 × 7		2 × 2 average pool, stride 2		
Dense Block (4)	7 × 7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1 × 1		7 × 7 global average pool		
			1000D fully-connected, softmax		

Table 3.1: Information about different versions of DenseNet

https://pytorch.org/hub/pytorch_vision_densenet/

The number of parameters rises as the number of layers grows, increasing the model's complexity. DenseNet is one of the most accurate medical imaging models, classifying breast histology images as benign or cancerous with a 94-95 percent accuracy [21]. Strong gradient flow, parameter and computational economy, more varied features, and low complexity features are all advantages of DenseNet.

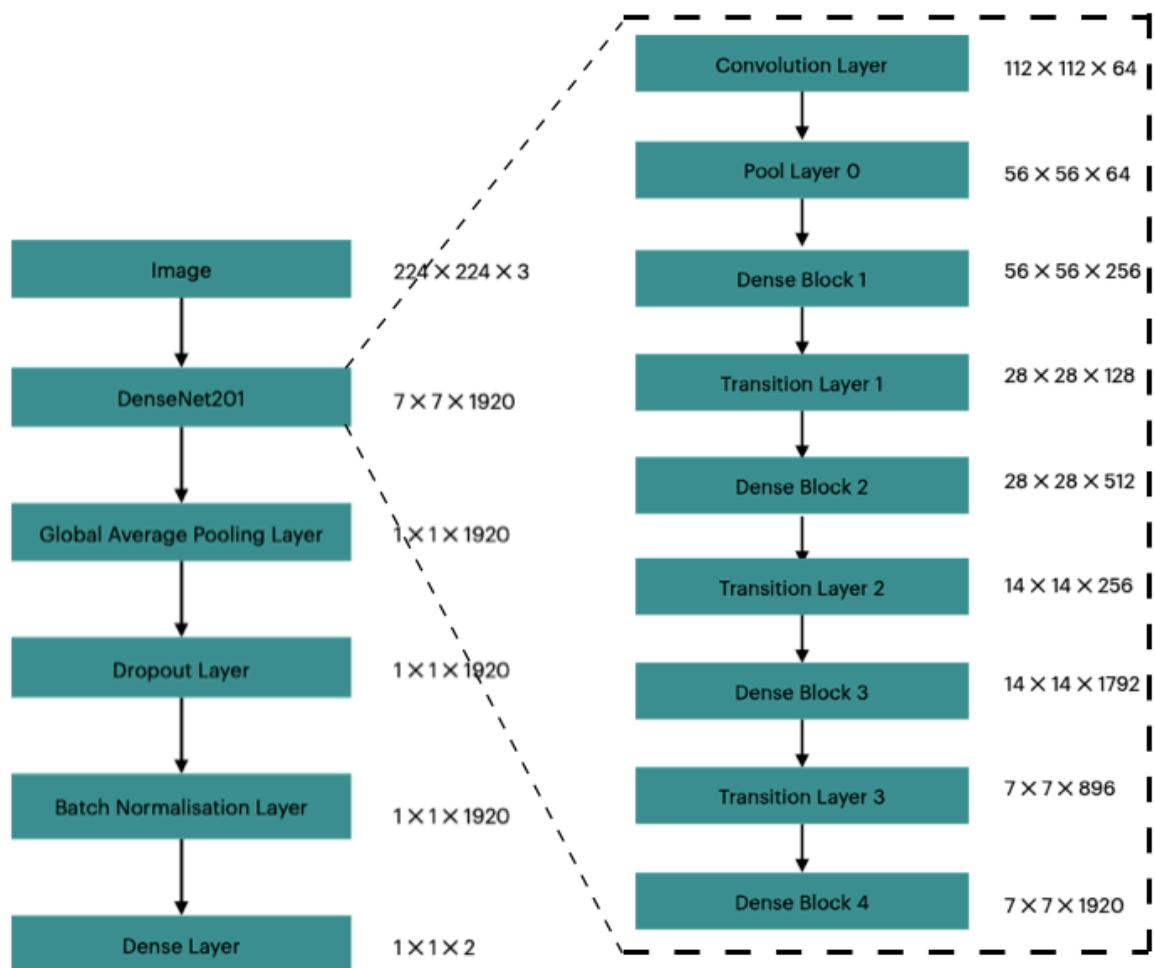


Figure 3.1: Structure of DenseNet

3.2 ResNets

ResNet is an abbreviation for Residual Network. In their publication "Deep Residual Learning for Image Recognition," Kaiming He, Shaoqing Ren, Xiangyu Zhang, and Jian Sun introduced a special sort of neural network in 2015. ResNets come in a variety of variants, including ResNeXt. ResNets come in a variety of variants, each with a distinct amount of layers. ResNet's skip connections alleviate the problem of disappearing gradient in deep neural networks by allowing the gradient to flow along an additional shortcut channel. These connections also aid the model by allowing it to learn the identity functions, ensuring that the higher layer performs at least as well as the lower layer, if not

better. The ResNet is the short for Residual Networks. It has many advantages like it can be trained easily while having a large number of parameters and it helps in solving the problem of vanishing gradient. It has a very good accuracy on the breast histology datasets. The resnets can classify the images with an accuracy of 94-95% [22] and hence it is used in many medical imaging techniques.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112			7×7, 64, stride 2		
conv2_x	56×56			3×3 max pool, stride 2		
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
	1×1			average pool, 1000-d fc, softmax		
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Table 3.2: Information about different versions of DenseNet
https://pytorch.org/hub/pytorch_vision_resnet/

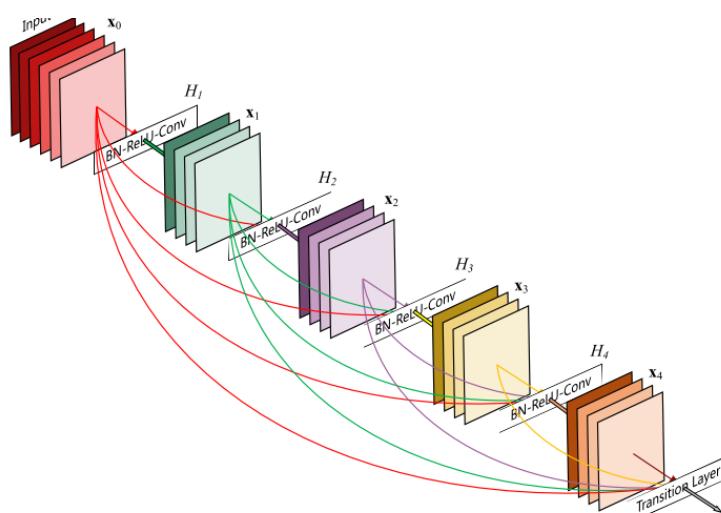


Figure 3.2: Structure of ResNets
<https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035>

3.3 VGG

It is named after the group that invented it that is Visual Geometry Group focused on computer vision at Oxford University. It has certain convolution layers and certain max-pooling layers stacked in a particular fashion and it has achieved many good results on many datasets. It has many advantages like multiple non-linear rectification layers instead of single layer. It helps to decrease the numbers of parameters while keeping same performance. There are some versions of VGG available like VGG13, VGG16, VGG19, etc. VGG19 was trained for the same dataset of breast histology and it achieved the result of 93-94%. The number of layers in VGG is less as compared to other models like DenseNet and ResNet. Still it is one of the best performing convolutional neural network used all over the world.

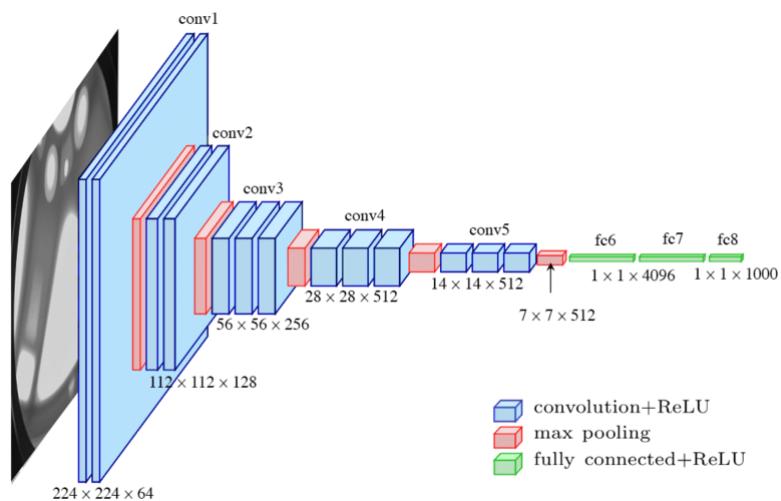


Figure 3.3: Structure of VGG16

https://www.researchgate.net/figure/Fig-A1-The-standard-VGG-16-network-architecture-as-proposed-in-32-Note-that-only_fig3_322512435

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 3.3: Information about different versions of VGG
<https://www.pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/>

3.4 Xception

Xception is the extreme version of Inception, with several unique elements not found in other versions of the game. It makes use of modified depth wise separable convolution and depth wise separable convolution. When compared to models that use either ReLU or ELU, the Xception model has the highest accuracy when there is no intermediate activation. The order in which the operations are done and the presence of non-linearity in the first operation are the two small differences in Xception. It uses depth wise convolution, which employs a 2D filter on each image layer.

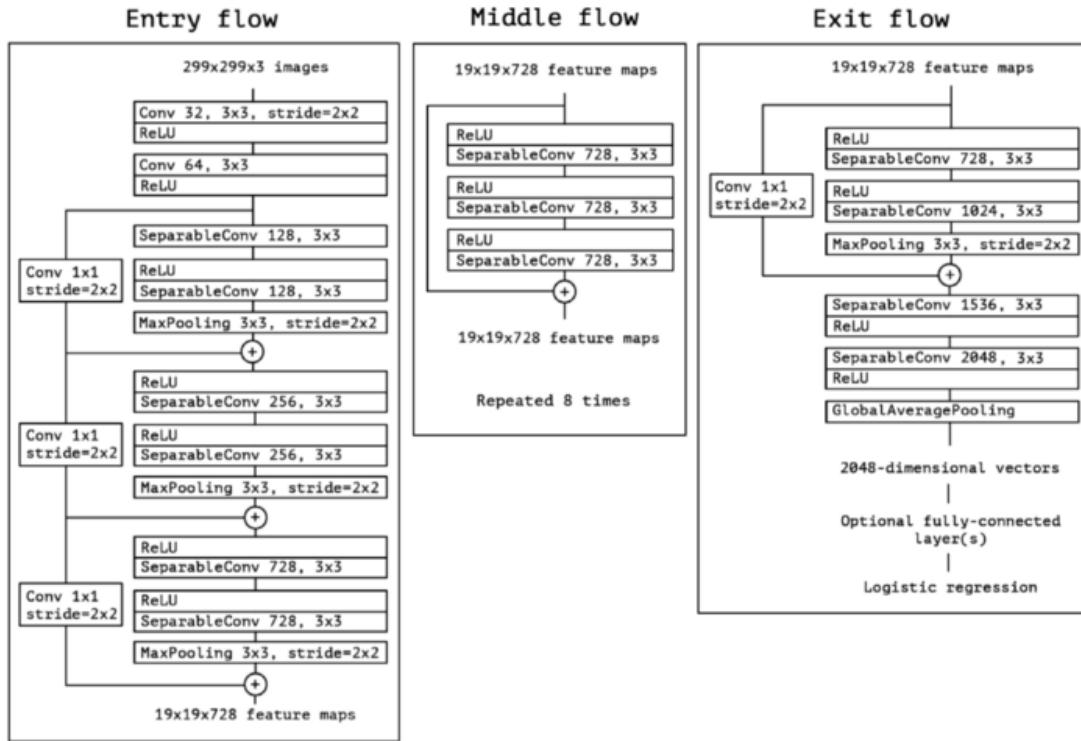


Figure 3.4: Overall Architecture of Xception network

<https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568>

3.5 Inception

Inception networks are made with complex parameters and it uses some tricks to further increase the performance; both in terms of accuracy and speed. It has 4 main types. These are Inception_v1, Inception_v2, Inception_v3, Inception_v4 and Inception_ResNet. Inception_v3 was used to train on the dataset and the testing accuracy was between 93-94%. Inception_v3 uses factorised 7x7 convolutions, auxiliary classifiers and label smoothening to facilitate the learning and increasing the accuracy.

These are the 5 models which were tested with the dataset available for the breast histology images.

type	patch size/stride or remarks	input size
conv	$3 \times 3 / 2$	$299 \times 299 \times 3$
conv	$3 \times 3 / 1$	$149 \times 149 \times 32$
conv padded	$3 \times 3 / 1$	$147 \times 147 \times 32$
pool	$3 \times 3 / 2$	$147 \times 147 \times 64$
conv	$3 \times 3 / 1$	$73 \times 73 \times 64$
conv	$3 \times 3 / 2$	$71 \times 71 \times 80$
conv	$3 \times 3 / 1$	$35 \times 35 \times 192$
$3 \times$ Inception	As in figure 5	$35 \times 35 \times 288$
$5 \times$ Inception	As in figure 6	$17 \times 17 \times 768$
$2 \times$ Inception	As in figure 7	$8 \times 8 \times 1280$
pool	8×8	$8 \times 8 \times 2048$
linear	logits	$1 \times 1 \times 2048$
softmax	classifier	$1 \times 1 \times 1000$

Table 3.4: Information about layers of Inception_v3

https://pytorch.org/hub/pytorch_vision_inception_v3/

Chapter 4

Methodology

4.1 Overview

All 5 models are trained and the accuracies were in the range of 93-95%. This indicates that there are some features which are not extracted by the models. There doesn't exist a single model which could extract all the important features required for very high accuracy classification. This gives an intuition of construction of a model which could extract most of the features required for the classification.

4.2 Abstract

It can be deduced that every model is capable of extracting features but misses some features. Different models can extract different features and can miss different features. The idea is to combine all the features extracted from the 5 models and then passing it to the classification module. This reduces the chance of missing any important feature because if a feature is not

extracted from model A then there is a high probability that the missing feature will be extracted from any other four models. Hence the complexity of the model will increase as the features increases but the accuracy will increase because most of the required features are present in the classification module.

4.3 Model Architecture

The training images will be fed into all 5 models (DenseNet201, ResNet152, VGG19, Xception, Inception_v3). The features extracted from the models and passed from a batch normalisation layer which is then concatenated and then fed into the classification module. The classification module contains 3 dense layers and 2 dropout layers. The last layer is a dense layer with the output dimension as a 1D vector of size 2. The first row contains the probability for the image to be malignant and the second row contains the probability for the image to be benign. The loss used for back propagation is binary cross entropy. We used Adam optimiser for the optimiser and softmax function for activation in the last dense layer.

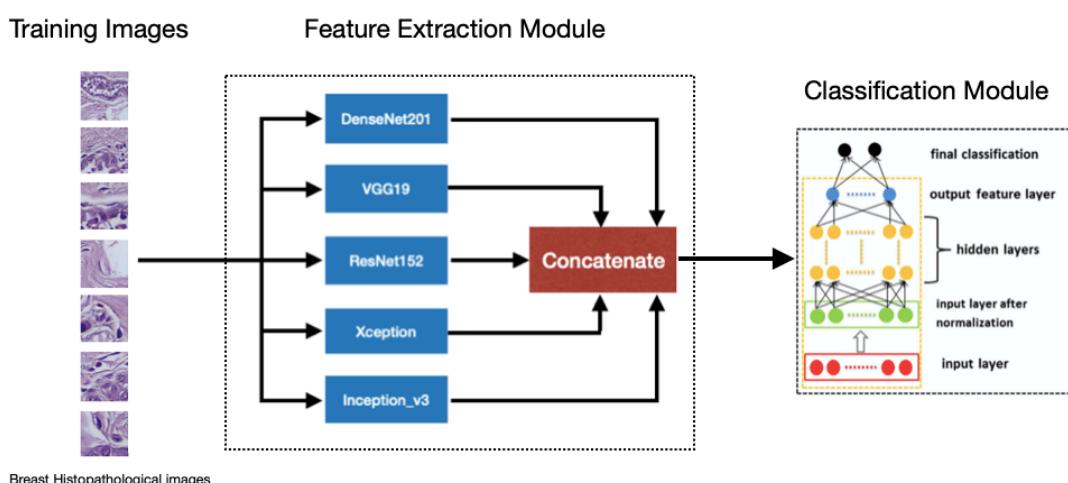


Figure 4.1: Flowchart for the proposed model

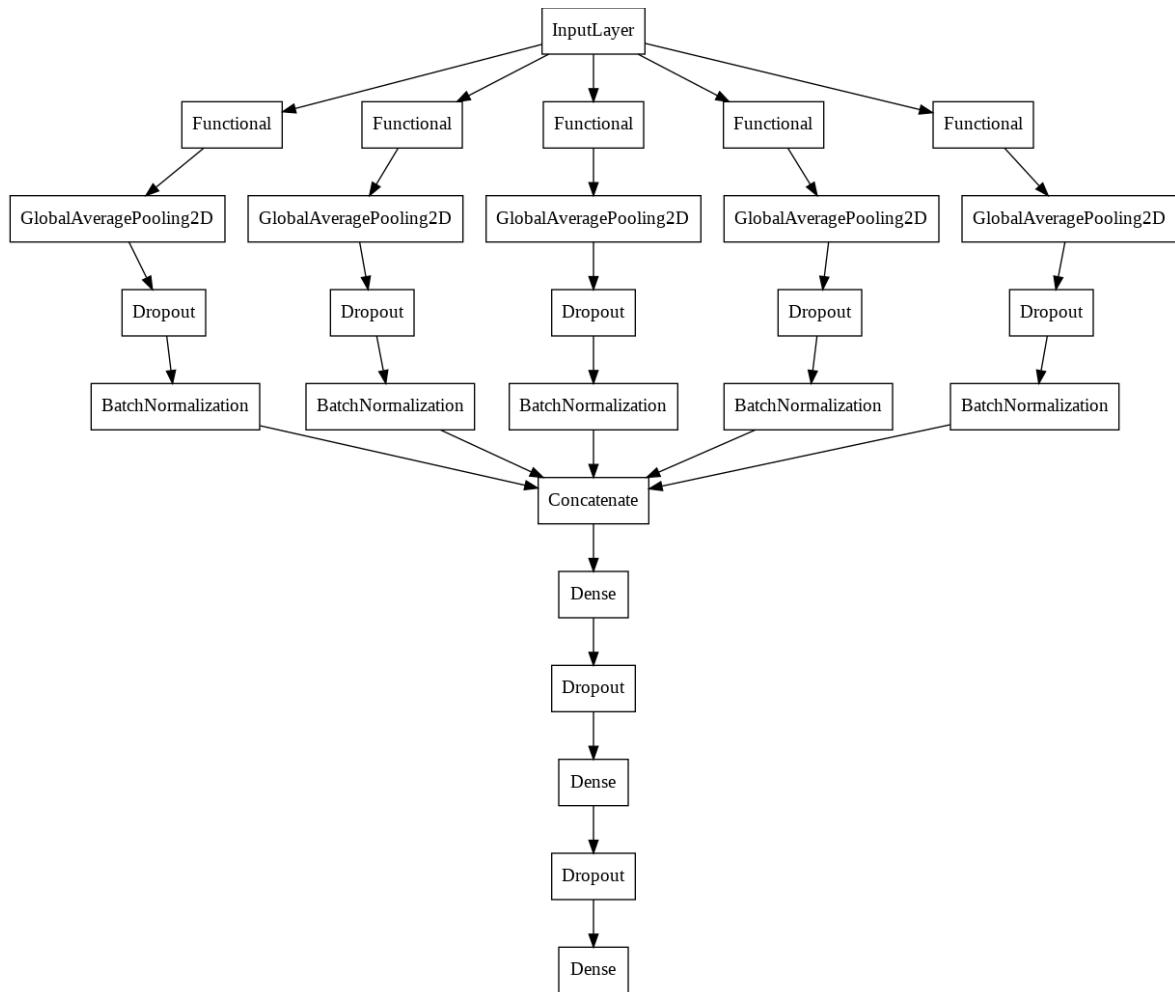


Figure 4.2: Architecture of the proposed model as plotted

The detailed diagram about the architecture can be found in the following google drive link:

<https://drive.google.com/file/d/1yRbtOLYErYgGAo9gXUDIUTBME4Js9rN3/view?usp=sharing>

4.3.1 Input Details

The images from the datasets are fed into the proposed model. The dimension of the image is $224 \times 224 \times 3$. These are coloured images with RGB format and the depth of the image varies with the datasets. The images are 8-bit depth for Breakhis dataset. The size images from Breakhis dataset ranges from 400kB to 700kB. The size of Kaggle BHI dataset ranges from 6kB to 7kB. The size of ICIAR2018 dataset varies from 18MB to 19MB.

4.3.2 Convolution Layers

This layer consists of neurons (or filters) that are connected to the input layer. A part of the input image is only connected to the convolution layer in order to reduce the computational complexity. The output of this layer is a feature map of the input image. The convolution operation is a dot product between the input image and the convolution layer filter.

$$f_c = conv(i, j) \quad (4.1)$$

$$= (I * K)(i, j) \quad (4.2)$$

$$= \sum_m \sum_n I(m, n) K(i - m, j - n) \quad (4.3)$$

Here, I is the input image, K is convolution kernel/filter, f_c is output feature map and $*$ denotes two dimensional discrete convolution operation.

4.3.3 Pooling

The pooling layer's goal is to downsample the feature map created by the convolution layer. The pooling layer reduces the computational cost of subsequent layers by condensing the feature map without sacrificing useful information. There are three different kinds of pooling. Maximum pooling, average pooling, and minimum pooling are

the three types of pooling. For example, in max pooling, a pooling unit produces the 2×2 sub-maximum matrix's activation value.

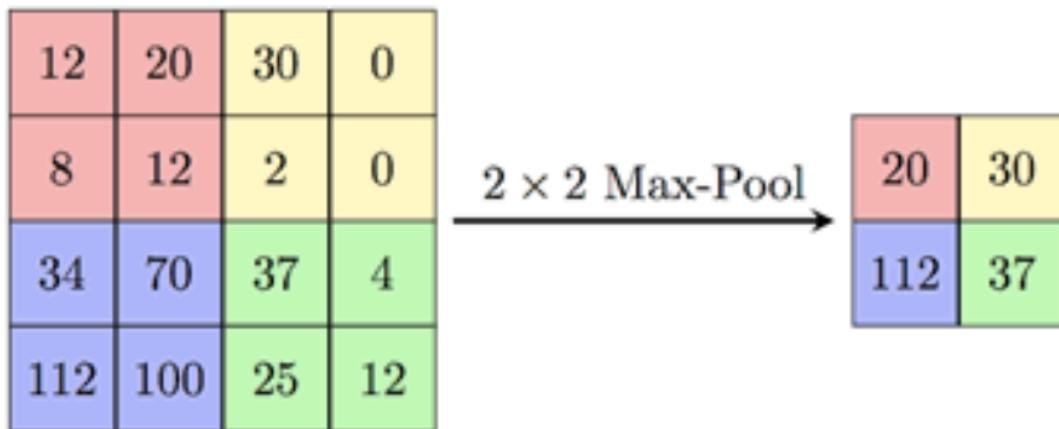


Figure 4.3: Max Pooling

4.3.4 Fully-connected Layer

Every neuron in the previous layer is connected to every neuron in the next layer in the completely connected layer, which is a multi-layer perceptron. The softmax activation feature is used in this layer. This layer uses the convolution layer's high-level features and pooling to classify the image into different groups based on the training dataset.

4.3.5 Dense Layer

A dense layer represents a matrix vector multiplication. The back propagation updates the trainable parameters in the matrix of the dense layer.

$$u^T \cdot W, \quad W \in R^{n \times m} \quad (4.4)$$

So it gets a m dimensional vector as output. Thus, a dense layer changes the dimension of the vector. It can be applied anywhere between any layers to match the dimension of 2 layers. It is also applied at the end of the model to match the dimension of the output. It applies a rotation, scaling, and translation transform to the vector mathematically..

4.3.6 Dropout Layer

A dropout layer is used for regularisation where we randomly set some of the dimensions of the input vector to be zero with probability. A dropout layer does not have any trainable parameters i.e. nothing gets updated during backward pass of back propagation. Dropout layer randomly removes the percentage of parameters from the available parameters to decrease the complexity of the model. We can provide the percentage of parameters we want to drop in the layers. To ensure that expected sum of vectors fed to this layer remains the same if no dropout was applied, the remaining dimensions which are not set to zero are scaled by:

$$\frac{1}{keep_prob} \quad (4.5)$$

4.3.7 Global Average Pooling Layer

To replace totally interconnected layers in a CNN, a pooling method known as global average pooling is performed. In the previous convolution layer, it builds a feature map for each of the classification task's categories. Rather than putting completely connected layers on top of the feature maps, it takes the average of each feature map and feeds the resulting vector directly into the activation layer.

4.3.8 Batch Normalisation Layer

Batch normalisation is a network layer that enables each layer to learn more independently. It's used to make the output of the previous layers more normal. In general, the mean of a matrix's values is set to around 0, and the range is set to [-1,1] or [0,1]. It makes learning easier because the ranges of different matrices are no longer distinct. The activations normalize the input layer by scaling it. Learning becomes more efficient when batch normalisation is performed, and it can also be used as a regularization to avoid model overfitting. To standardize the input or outputs, the layer is added to the sequential model. It can be put between the layers at various points in the model. It's commonly inserted after the sequential model has been set up, but before the convolution and pooling layers.

4.3.9 Concatenation Layer

The concatenation layer receives a list of inputs and combines them. Technically it takes a list of tensors as the input and combines them to form a single tensor. In the proposed model, the concatenation layer is used to combine the extracted features from different models and combining them to form a single tensor of features.

4.4 Layers in the model

The convolution networks are stacked with batch normalisation layers and dropout layers before feeding into the concatenation layer. The input is in the form of an image of dimension $224 \times 224 \times 3$ and it is fed into all 5 models. The number of parameters in DenseNet201, VGG19, ResNet152, Xception and Inception_v3 are 18,321,984, 20,024,384, 58,370,944, 20,861,480 and 21,802,784 respectively. The total number of trainable parameters in the model is 148,509,290. The model comprises of different layers including the 5 networks as explained in the previous section, global average pooling layers, dropout layers, dense layers and batch normalisation layers. The output of the model is a vector of size 2 which contains the probability of the image to lie in one of the classes.

Model: "model"				
Layer (type)	Output Shape	Param #	Connected to	
input_1 (InputLayer)	[None, 224, 224, 3] 0			
densenet201 (Functional)	(None, 7, 7, 1920)	18321984	input_1[0][0]	
vgg19 (Functional)	(None, 7, 7, 512)	20024384	input_1[0][0]	
resnet152 (Functional)	(None, 7, 7, 2048)	58370944	input_1[0][0]	
xception (Functional)	(None, 7, 7, 2048)	20861480	input_1[0][0]	
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1920)	0	densenet201[0][0]	
global_average_pooling2d_3 (GlobalAveragePooling2D)	(None, 512)	0	vgg19[0][0]	
global_average_pooling2d_4 (GlobalAveragePooling2D)	(None, 2048)	0	resnet152[0][0]	
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 2048)	0	xception[0][0]	
inception_v3 (Functional)	(None, 2048)	21802784	input_1[0][0]	
dropout (Dropout)	(None, 1920)	0	global_average_pooling2d[0][0]	
dropout_3 (Dropout)	(None, 512)	0	global_average_pooling2d_3[0][0]	
dropout_4 (Dropout)	(None, 2048)	0	global_average_pooling2d_4[0][0]	
dropout_1 (Dropout)	(None, 2048)	0	global_average_pooling2d_1[0][0]	
dropout_2 (Dropout)	(None, 2048)	0	inception_v3[0][0]	
batch_normalization (BatchNormalization)	(None, 1920)	7680	dropout[0][0]	
batch_normalization_101 (BatchNormalization)	(None, 512)	2048	dropout_3[0][0]	
batch_normalization_102 (BatchNormalization)	(None, 2048)	8192	dropout_4[0][0]	
batch_normalization_5 (BatchNormalization)	(None, 2048)	8192	dropout_1[0][0]	
batch_normalization_100 (BatchNormalization)	(None, 2048)	8192	dropout_2[0][0]	
concatenate_2 (Concatenate)	(None, 8576)	0	batch_normalization[0][0] batch_normalization_101[0][0] batch_normalization_102[0][0] batch_normalization_5[0][0] batch_normalization_100[0][0]	
dense (Dense)	(None, 1000)	8577000	concatenate_2[0][0]	
dropout_5 (Dropout)	(None, 1000)	0	dense[0][0]	
dense_1 (Dense)	(None, 1000)	1001000	dropout_5[0][0]	
dropout_6 (Dropout)	(None, 1000)	0	dense_1[0][0]	
dense_2 (Dense)	(None, 2)	2002	dropout_6[0][0]	

Total params: 148,995,882
Trainable params: 148,509,290
Non-trainable params: 486,592

Figure 4.3: Information about the layers of the model

Chapter 5

Implementation and Results

The model is trained for all the 3 datasets and tested, the accuracy graphs, validation accuracy graphs, loss, validation loss are plotted and the model results is compared to different existing models. The model is trained on Google Colab as it required a GPU.

5.1 Training

The dataset is divided in the ratio of 80:20 for training and testing. It is divided manually. A batch size of 16 is used and epoch of 100 is used to train the dataset. A learning rate of 0.0001 is used. The Breakhis dataset [9] took 32 hours to train, the Kaggle BHI [10] took 10 hours to train and ICIAR2018 dataset [11] took 2 hours to train. Since the Breakhis dataset [9] is distributed into four different divisions as the magnification factors of 40x, 100x, 200x and 400x. The testing is done separately for all 4 magnifications to get the individual results. The Breakhis dataset [9] contains a total of 7909 images, the ICIAR2018 contains a total of 400 images before data augmentation and the Kaggle BHI contains 2401 images. The following table shows number of images used for training and testing for all the 3 datasets.

Dataset Name		TRAIN		TEST	
		Benign	Malignant	Benign	Malignant
<i>Breakhis</i>	40x	501	1097	126	275
	100x	517	1151	129	288
	200x	500	1113	125	279
	400x	472	987	118	247
ICCIAR 2018		652	712	148	88
KAGGLE BHI		1149	791	280	181

Table 5.1: Information about the number of training and testing images for Breakhis, ICIAR2018 and Kaggle BHI dataset

5.2 Testing

20% of the images are used for testing purposes. Figure 5.1 contains the information about the number of images used in training and testing for all three datasets. The splitting of dataset into training and testing samples is done manually. The ICIAR2018 dataset contains only 400 images, hence the model could not work properly for this dataset due to the limitation of images, hence a data augmentation technique is used in which the number of images is increased to 4 times by inserting different rotations of the same image. These were 90°, 180° and 270° rotations.

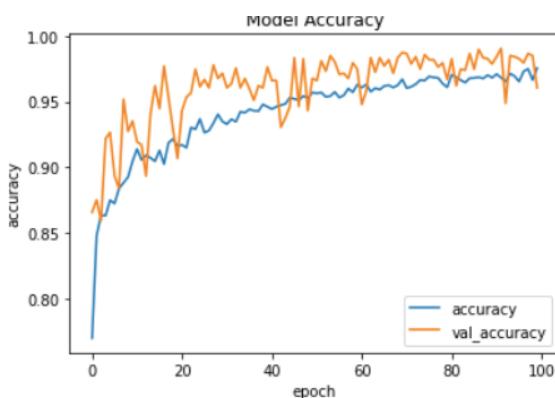


Figure 5.1: Accuracy and validation accuracy for 100 epochs training of the model

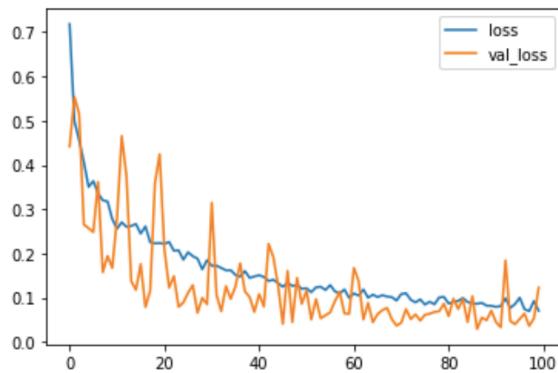


Figure 5.2: Loss and validation loss for 100 epochs training of the model

5.3 Experimental Results

The model is tested for the testing dataset and certain performance quantities are calculated for each dataset. A performance matrix is also calculated to find out the sensitivity (recall), specificity, precision and the F-score. For any classification algorithm, the efficiency is calculated using these numbers. Hence all the quantities are calculated for all the 3 datasets, including 4 different magnifications of BreakHis dataset.

5.3.1 Performances quantities

All the calculations are made considering the malignant images to be positive. The performance metrics contains 4 quantities which are shown in the following equations. These are True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN).

True Positive(TP): The number of positive observations that were actually positive, as predicted by the model.

True Negative(TN): The number of negative observations that were actually negative, as predicted by the model.

False Positive(FP): The number of positive observations predicted by the model that were actually negative.

False Negative(FN): The number of negative observations predicted by the model that were actually positive.

The quantities Accuracy, Precision, Recall, Specificity, Sensitivity and F1 score calculations in the following equations.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + FN + TN)} \quad (5.1)$$

$$\text{Recall} = \frac{TP}{(TP + FN)} \quad (5.2)$$

$$\text{Precision} = \frac{TP}{(TP + FP)} \quad (5.3)$$

$$\text{Specificity} = \frac{TN}{(TN + FP)} \quad (5.4)$$

$$\text{Sensitivity} = \frac{TP}{(TP + FN)} \quad (5.5)$$

$$\text{F1 - Score} = 2 * \frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \quad (5.6)$$

5.3.2 Accuracy of the Trained Network

The accuracy is obtained and all the performance values are calculated which are summarised in Table 5.2 for all 3 dataset.

<i>Dataset Name</i>	<i>Accuracy</i>	<i>Sensitivity/ Recall</i>	<i>Specificity</i>	<i>Precision</i>	<i>F1-score</i>
<i>Breakhis</i>	40x	95.98%	98.49%	90.97%	95.62% 0.97
	100x	98.31%	98.61%	97.63%	98.95% 0.98
	200x	97.51%	97.18%	98.30%	99.28% 0.98
	400x	94.21%	94.82%	92.85%	96.74% 0.95
<i>ICIAR 2018</i>	91.52%	81.48%	100%	100%	0.90
<i>KAGGLE BHI</i>	96.74%	99.40%	95.22%	92.26%	0.96

Table 5.2: Information about the performance values (accuracy, sensitivity (recall), specificity, precision and F1-score for different datasets as obtained from the proposed model

As it can be seen from Table 5.2 the accuracy for Breakhis dataset is promising for all magnifications except 400x. The model does not perform well on ICIAR2018 dataset because of the limited number of images in the dataset. The model also has good accuracy for the Kaggle BHI dataset.

5.3.3 Confusion Matrix

The confusion matrix for all the dataset and magnification values are plotted and are provided in figure 5.3 and Figure 5.4.

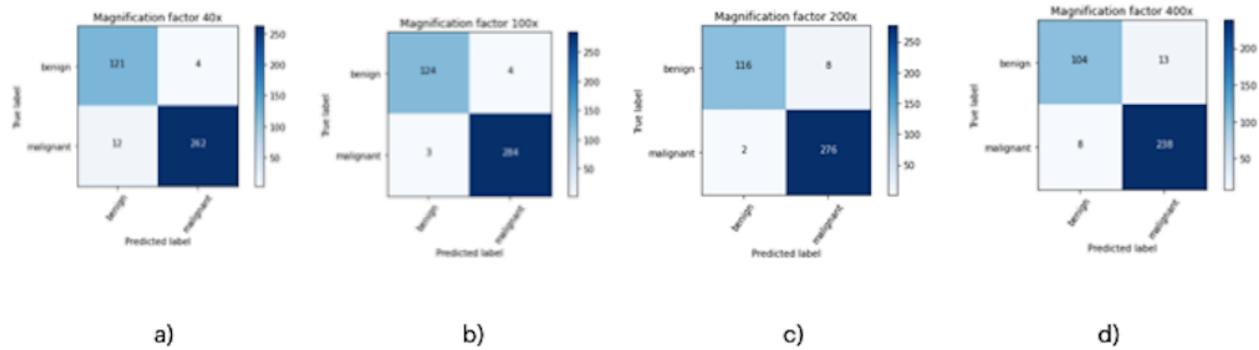


Figure 5.3: Confusion Matrix for different magnification factors for Breakhis dataset
a) 40x b) 100x c) 200x d) 400x

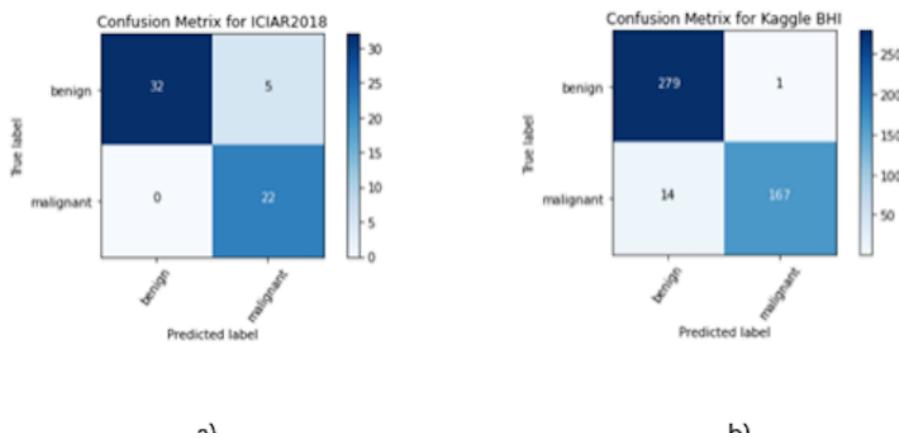


Figure 5.4: Confusion Matrix for a) ICIAR2018 dataset b) Kaggle BHI dataset

Chapter 6

Conclusions

By training different magnifications of images to learn the general structures and texture aspects of cells, I suggested a model that incorporates the attributes gathered from 5 separate pre-existing models for categorization of histology pictures. The model accuracies and other measurable quantities were tested on three publicly available standard datasets, Breakhis ICIAR2018 and Kaggle BHI, which produced good results that were similar to all other models available. The suggested model's accuracies are compared to those of other works in Table 6.1.

References	Methods	40x	100x	200x	400x
Kahya et al. (2017) [12]	ASSVM	94.97	93.62	94.54	94.42
Wei et al. (2017) [13]	BiCNN	97.89	97.64	97.50	97.97
Pratiher et al. (2018) [14]	L-Isomap and SSAE	96.8	98.1	98.2	97.5
Bardou et al. (2018) [15]	CNN + Augmented	96.82	96.96	96.36	95.97
Bardou et al. (2018) [15]	SVM	92.71	93.75	92.72	92.12
Present Work	Parallel CNN	95.98	98.31	97.51	94.21

Table 6.1: Comparison of the proposed model with other existing models

The model is able to get good results and it is totally comparable to some previous classification models in certain performance values such as accuracy, sensitivity, and specificity. As compared to different models in the previous table, it can be seen then the present work is performing quite good as compared to other algorithms. It is surely performing better on the 100x and 200x as it achieved 98.31% and 97.51% on 100x and 200x respectively. For the ICIAR2018 dataset, the model didn't achieve good accuracy because of less number of images to train. The ICIAR2018 dataset [11] contains only 400 images and the model did not fit well for such a small dataset. Although after increasing the number of images by using data augmentation.

The training data was increased 4 times and hence the accuracy increased by approximately 1%. The proposed model also achieved good accuracy of 94.14% on Kaggle Breast Histology Image dataset [10]. The Kaggle BHI dataset is very big as it contains 2,77,524 images hence training on the full dataset is not possible on a CPU, therefore a subset of the dataset was used for training and testing purposes. The suggested model demonstrated great sensitivity and specificity for a variety of instances, making it beneficial for pathologists and researchers working in the field of cancer diagnosis utilizing histological pictures. The idea of combining the features from different models has increased the accuracy as compared to independent models and this technique can be applied on other algorithms as well.

References

1. Abdullah-Al Nahid; Mohamad Ali Mehrabi; Yinan Kong, Histopathological Breast Cancer Image Classification by Deep Neural Network Techniques Guided by Local Clustering, BioMed Research International 2018. <https://doi.org/10.1155/2018/2362108>
2. Spanhol; Oliveira; Petitjean; Heutte: Breast cancer histopathological image classification using convolutional neural networks. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; pp. 2560–2567.
3. Ghulam Murtaza¹; Liyana Shuib¹; Ainuddin Wahid Abdul Wahab¹; Ghulam Mujtaba²; Ghulam Mujtaba³; Ghulam Raza⁴ and Nor Aniza Azmi⁵: Breast cancer classification using digital biopsy histopathology images through transfer learning. Journal of Physics: Conference Series, Volume 1339, International Conference Computer Science and Engineering (IC2SE) 26–27 April 2019, Padang, Indonesia, Ghulam Murtaza et al 2019 J. Phys.: Conf. Ser. 1339 012035
4. Murtaza, G., Shuib, L., Mujtaba, G. et al. Breast Cancer Multi-classification through Deep Neural Network and Hierarchical Classification Approach. *Multimed Tools Appl* 79, 15481–15511 (2020). <https://doi.org/10.1007/s11042-019-7525-4>
5. Murtaza, G., Shuib, L., Abdul Wahab, A.W. et al. Deep learning-based breast cancer classification through medical imaging modalities: state of the art and research challenges. *Artif Intell Rev* 53, 1655–1720 (2020). <https://doi.org/10.1007/s10462-019-09716-5>
6. Assiri, A.S.; Nazir, S.; Velastin, S.A. Breast Tumor Classification Using an Ensemble Machine Learning Method. *J. Imaging* 2020, 6, 39. <https://doi.org/10.3390/jimaging6060039>

7. Chaurasia V, Pal S, Tiwari B. Prediction of benign and malignant breast cancer using data mining techniques. *Journal of Algorithms & Computational Technology*. June 2018;119-126. doi:10.1177/1748301818756225
8. Sheikh, T.S.; Lee, Y.; Cho, M. Histopathological Classification of Breast Cancer Images Using a Multi-Scale Input and Multi-Feature Network. *Cancers* 2020, 12, 2031. <https://doi.org/10.3390/cancers12082031>
9. Spanhol, F., Oliveira, L. S., Petitjean, C., Heutte, L., A Dataset for Breast Cancer Histopathological Image Classification, *IEEE Transactions on Biomedical Engineering (TBME)*, 63(7):1455-1462, 2016.
10. Paul Mooney; kaggle.com: Breast Histopathology Images, 198,738 IDC(-) image patches; 78,786 IDC(+) image patches, 2018.
11. Guilherme Aresta, Daniel Riccio, Yaqi Wang, Lingling Sun, Kaiqiang Ma, Jiannan Fang, Ismael Kone, Lahsen Boulmane, Aurélio Campilho, Catarina Eloy, António Polónia, Paulo Aguiar, BACH: Grand challenge on breast cancer histology images, *Medical Image Analysis*, 2019, ISSN 1361-8415, <https://doi.org/10.1016/j.media.2019.05.010>.
12. Kahya MA, Al-Hayani W, Algamal ZY. Classification of breast cancer histopathology images based on adaptive sparse support vector machine. *Journal of Applied Mathematics and Bioinformatics*. 2017; 7 (1):49.
13. Wei B, Han Z, He X, Yin Y. Deep learning model based breast cancer histopathological image classification. In: Cloud Computing and Big Data Analysis (ICCCBDA), 2017 IEEE 2nd International Conference on. IEEE; 2017.
14. Pratiher S, Chattoraj S. Manifold Learning & Stacked Sparse Autoencoder for Robust Breast Cancer Classification from Histopathological Images. *arXiv preprint arXiv:180606876*. 2018;
15. Bardou D, Zhang K, Ahmad SM. Classification of Breast Cancer Based on Histology Images Using Convolutional Neural Networks. *IEEE Access*;PP(99):1–1.

16. Y. Jiang, L. Chen, H. Zhang, and X. Xiao, Breast cancer histopathological image classification using convolutional neural networks with small seresnet module. PloS one, vol. 14, no. 3, p. e0214587, 2019
17. Hongdou Yao, Xuejie Zhang , Xiaobing Zhou, Shengyan Liu, Parallel Structure Deep Neural Network Using CNN and RNN with an Attention Mechanism for Breast Cancer Histology Image Classification , 2019, <https://www.mdpi.com/2072-6694/11/12/1901>
18. Bo-Yao Lin, Chu-Song Chen; Two Parallel Deep Convolutional Neural Networks for Pedestrian Detection, 2015 IEEE
19. Jiang, Y.; Chen, L.; Zhang, H.; Xiao, X. Breast cancer histopathological image classification using convolutional neural networks with small SE-ResNet module. PLoS ONE 2019, 20, e0214587.
20. Gurcan, M.N.; Boucheron, L.; Can, A.; Madabhushi, A.; Rajpoot, N.; Yener, B. Histopathological image analysis: A review. IEEE Rev. Biomed. Eng. 2009, 2, 147.
21. Yuliana Jiménez Gaona, María José Rodriguez-Alvarez, University of Waterloo, Waterloo, ON N2L3G1, Canada; DenseNet for Breast Tumor Classification in Mammographic Images
22. Q. A. Al-Haija and A. Adebanjo, "Breast Cancer Diagnosis in Histopathological Images Using ResNet-50 Convolutional Neural Network," 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 2020, pp. 1-7, doi: 10.1109/IEMTRONICS51293.2020.9216455.

Web References

1. <https://towardsdatascience.com/>
2. <https://medium.com/>
3. <https://www.analyticsvidhya.com/>
4. <https://www.kaggle.com/>
5. <https://en.wikipedia.org/>
6. <https://machinelearningmastery.com/>
7. <https://www.coursera.org/>
8. <https://www.deeplearning.ai/>
9. <https://www.geeksforgeeks.org/>
10. <https://pytorch.org/>
11. <https://www.healthline.com/>
12. <https://doi.org/10.1371/>
13. <https://www.researchgate.com/>
14. <https://www.pimagesearch.com/>