Invasive Ductal Carcinoma Classification using CNNs in Breast Histopathological Images

A thesis submitted in fulfilment of the requirements for the degree of Bachelor of Technology

by

Kushal Deb, CSE Department

Prem Chand Panwar, ECE Department

Archana MD, ECE Department

Under the supervision of

Dr. K T Deepak, Asst. Professor, ECE Department

Dr. Kavi Mahesh, Director

Dharwad

ज्ञानेन विकासः

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DHARWAD

May 2019

CERTIFICATE

It is certified that the work contained in this thesis entitled Invasive Ductal Carcinoma Classification using CNNs in Breast Histopathological Images by Kushal Deb, Prem Chand Panwar and Archana MD has been carried out under our supervision and that it has not been submitted elsewhere for a degree.

Dr. K T Deepak
Asst Professor
ECE Department

Dr Kavi Mahesh
Director

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission.
We 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.

Kushal Deb, CSE Department
Prem Chand Panwar, ECE Department
Archana M D, ECE Department

APPROVAL SHEET

This project report entitled Invasive Ductal Carcinoma Classification using CNNs in Breast Histopathological Images by Kushal Deb (15CS06) is approved for the degree of Bachelor of Technology in Computer Science and Engineering.

Supervisors:

Dr. K T Deepak	Dr Kavi Mahesh
Asst Professor	Director
ECE Department	

Head of Department:

Dr Arun Chauhan

CSE Department

Abstract

Breast cancer disease is a disease that happens when cells in breast tissue change (or transform or mutate) and continue imitating (or reproducing). These unusual cells for the most part bunch together to form a tumor. Invasive ductal carcinoma (IDC), otherwise called infiltrating ductal carcinoma is the most common form of breast cancer, speaking to 80 percent of all breast cancer diagnoses. It is a cancer that starts developing in a milk duct and attacks the fibrous or fatty tissue of the breast outside of the duct.

Globally, breast cancer is the most common disease among women, after skin cancer. It is likewise the second leading reason for cancer death in women after lung disease. Breast cancer ranks as the number one cancer among Indian females with rate as high as 25.8 per 1,00,000 women and mortality of 12.7 per 1,00,000 women according to the health ministry.

In this project we will classify IDC and Non-IDC cells from Breast Histopathological Images using various state-of-the-art Convolutional Neural Network (CNN) architectures namely LeNet, AlexNet, ZFNet, VGG16, VGG19, Inception, InceptionResnet, Densenet, Xception and Mobilenet. The dataset contains 198,738 Non-IDC images and 78,786 IDC images collected from 162 whole mount slide images of Breast Cancer (BCa) specimen scanned at 40x. The breast tissues are stained with an artificial dye. In this classification system IDC cells are classified as 1 and Non-IDC cells are classified as 0. Conventional methods often take more time in analysis. This system is not meant to be a stand-alone system but a means to assist doctors and pathology technicians to classify correctly faster and better. Proper and early classification of IDC cells will result in better mortality rate among women with breast cancer.

Acknowledgement

We wish to express our gratitude to Dr. Kavi Mahesh, Director, IIIT Dharwad and Dr. KT Deepak, Assistant Professor, ECE Department, IIIT Dharwad for their constant support and valuable guidance. We are really thankful to them for taking out time to guide us in the project inspite of their busy schedule. The project benefitted a lot from their invaluable guidance.

We also take this opportunity to thank Crowd Analytix, Bangalore to allow us to use the GPU machines in their facility. This enabled us to complete the project in the stipulated time.

Kushal Deb, CSE Department
Prem Chand Panwar, ECE Department
Archana M D, ECE Department

Contents

List of Figures

List of Tables

Chapter 1 Introduction

Cancer is a cluster or group of diseases including abnormal cell development with a possibility to attack and spread in different parts of the body. Human body is made up by the fundamental units called cells. Cells develop and grow to make new cells as the body needs them. As a rule, cells die when they get excessively old or harmed. At that point, new cells take their place. Cancer starts when genetic changes in the human body meddle with the deliberate procedure. Cells begin to grow and develop wildly. These cells may form a mass called tumor. A tumor can be malignant or benign. Cancer is the second driving reason for death globally, and is in charge of an expected 9.6 million passings in 2018. Globally, around 1 of every 6 passings is because of cancer. The financial effect of cancer is huge and is expanding. The absolute yearly financial expense of cancer in 2010 was assessed at roughly US$ 1.16 trillion. As indicated by National Cancer Registry Program of India (ICMR), in excess of 1300 Indians bite the dust each day because of disease. According to National Cancer Registry Programme of India (ICMR), the rate of mortality because of malignancy in India was high and disturbing with around 806000 existing cases by the end of the last century. Cancer is the second most common disease in India and responsible for the greatest mortality with about 0.3 percent deaths every year.

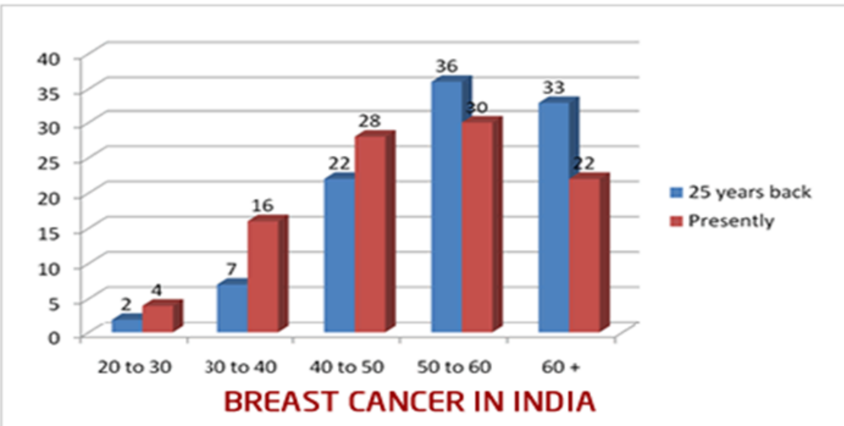

Figure 1: Increasing Incidence of Breast Cancer in Younger Age Groups

Figure Source: http://www.breastcancerindia.net/statistics/trends.html

As indicated by the Centers for Disease Control and Prevention (CDC), breast cancer is the most widely recognized cancer in women. Breast Cancer is the main source of death by disease in women matured somewhere in the range of 20 and 59 years and the second for women matured over 59 years. Breast cancer positions as the main disease among Indian females with rate as high as 25.8 per 1,00,000 women and mortality of 12.7 per 1,00,000 women as indicated by the health ministry. In India, we are currently seeing an ever increasing number of quantities of patients being determined to have breast cancer to be in the more younger age groups.

Invasive ductal carcinoma (IDC) is the most common form of all breast cancers. Invasive means that the cancer has a tendency to invade or spread to the surrounding breast tissues. Ductal means that the cancer began in the milk ducts, which are the pipes that carry milk from the milk-producing lobules to the nipple. About 80% of breast cancers are Invasive ductal carcinomas and overtime IDCs can spread to the lymph nodes and possibly to other areas. Early detection of IDCs can increase the life expectancy of a woman greatly.

The purpose of this work is to accurately identify and categorize breast cancer sub types as IDC or Non IDC and automate methods to save time and reduce error. The methodology involves training breast histopathological images using various state-of-the-art Convolutional Neural Network (CNN) architectures namely LeNet, AlexNet, ZFNet, VGG16, VGG19, Inception, InceptionResnet, Densenet, Xception and Mobilenet. Finally we used the top 5 best performing CNN architectures to create an ensemble model.

Chapter 2 Literature Survey

Breast cancer is the most common type of cancer among women, and image analysis methods that target the disease have an enormous potential to decrease the workload in a typical pathology lab and to improve the quality of the interpretation. It begins with an outline of the tissue preparation, staining and slide digitization processes followed by a discussion of the different image processing techniques and applications, ranging from analysis of tissue staining to computer-aided diagnosis, and forecast of breast cancer patients. [38]

We find that the convolutional neural networks architecture has not been used or not discussed in literature. So we aim to conduct a study on such art of architecture and train on these architectures. Latest deep learning approach dependent on a Convolutional Neural Network (CNN) model for multi-class breast cancer classification. The proposed methodology intends to characterize the breast tumors in non-just benign or malignant. Significantly Convolutional Neural Networks are utilized to straightforwardly characterize pre-segmented breast masses in mammograms as benign or malignant, utilizing a mix of transfer learning, cautious pre-preparing and information growth to overcome limited training data. [40]

Conventional classification approaches depend on feature extraction methods intended for a specific problem based on field-knowledge. To overcome the numerous challenges of the feature-based approaches, deep learning methods are becoming important alternatives. A method for the classification of hematoxylin and eosin stained breast biopsy images using Convolutional Neural Networks (CNNs) was proposed. [41]

Radiologists frequently experience serious difficulties classifying mammography mass lesions which leads to unnecessary breast biopsies to remove suspicions and this ends up adding exorbitant expenses to an already burdened patient and health care system. The breast cancer detection accuracy and efficiency can be increased by applying various image analysis techniques on digital mammograms on the dense regions of the breasts helping the radiologists to identify suspicious regions preventing unwanted biopsies and traumatic treatments. In 2018 Computer-aided Diagnosis (CAD) system based on deep Convolutional Neural Networks (CNN) that aims to help the radiologist classify mammography mass lesions was proposed. [42]

Regular check-ups are crucial for early detection and treatment of breast cancer type. Pathologist performs the diagnosis of the breast cancer. Recent computer-aided methods for breast cancer diagnosis permit another and faster way of breast cancer diagnosis. In 2017, Artificial neural networks trained by back propagation, a method for automatic classification of images for breast cancer diagnosis was proposed. The performance of the automatic classification of the breast cancer images was additionally improved by utilizing radial basis neural networks (RBFN). [43]

They proposed the Convolution neural networks on mammograms for detection of normal and abnormal mammograms. This deep learning technique is used on mammograms MIAS dataset by extracting features from sub-divided abnormal classes to the normal class. [44]

Specifically, a Convolutional Neural Network (CNN), a Long-Short-Term-Memory (LSTM), and a combination of CNN and LSTM are proposed for breast cancer image classification. Softmax and Support Vector Machine (SVM) layers have been used for the decision-making stage after extracting features utilising the proposed novel DNN models. [45]

Chapter 3 Dataset

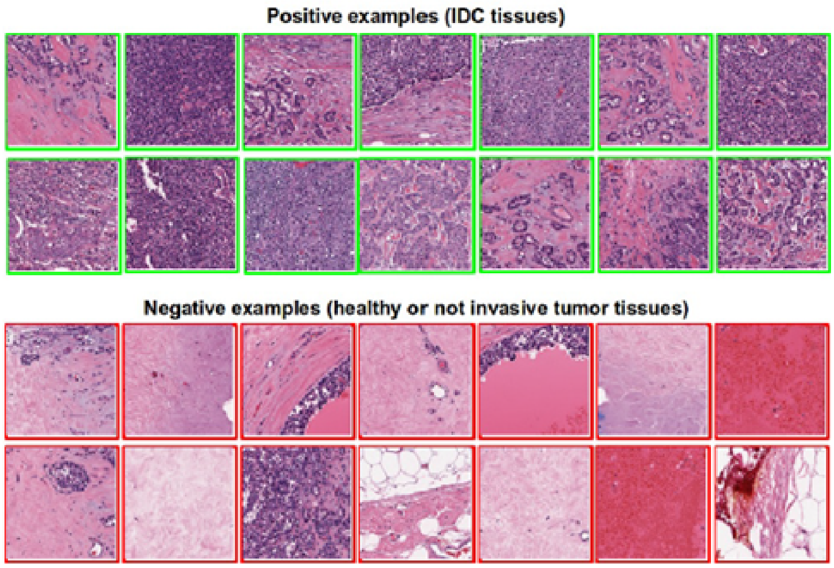

Figure 2: Whole Slide Images of IDC and Non-IDC

Figure Source: Automatic detection of invasive ductal carcinoma in whole slide images with Convolutional Neural Networks

Breast Cancer Histopathological Images is an openly available dataset found on Kaggle consisting of 277,524 images of which 198,738 are Non IDC and 78,786 are IDC images. Histopathology is the study of the signs of the disease using the microscopic examination of a biopsy or surgical specimen that is processed and fixed onto glass slides. To visualize different segments of the tissue under a microscope, the sections are dyed with one or more stains. Digital cameras are increasingly used to capture histopathology images. To assign an aggressiveness grade to a whole mount sample, pathologists typically focus on the regions which contain the IDC. As a result, one of the common pre-processing steps for automatic aggressiveness grading is to delineate the exact regions of IDC inside of a whole mount slide.

The original dataset consisted of 279 fully mount slide histopathological images of breast cancer specimens scanned at 40x. From these patches, images of dimensions 50x50 were extracted.

Histopathology is the microscopic examination of tissues in order to find indication of a disease. In medical terms, histopathology refers to biopsy or surgical specimen. The cells are dyed with artificial dyes and digital images of these tissues are captured.

The original dataset is in the form of patient IDs. The dataset contains 279 patient IDs each of which contains two folders for IDC and Non-IDC each. All the images for each of the patients is unique. We have used these images from various patients for the project.

The dataset contains a total of 277,524 images of which 198,738 images are IDC and rest 78,786 are Non-IDC. As we can see the dataset is imbalanced, we cannot use the original dataset for our training purposes. Moreover, due to computation limitations we cannot use all the images for our training and validation purposes. Hence, we decided to undersample the dataset. We used a total of 100,000 images for training the CNNs (with 50,000 images each for the two classes) and used a total of 10,000 images (5,000 image each for the two classes) for validation purposes. The images have been selected randomly to avoid any bias.

Chapter 4 Convolutional Neural Network

Convolutional Neural Network (CNN) is a class of deep learning algorithms, a specific type of artificial neural network that uses perceptrons, a machine learning unit for supervised learning to analyze data. A perceptron is the most fundamental unit of an artificial neural netrork which takes an input and returns an output after passing it through a activation function. CNNs are mainly used for Computer Vision tasks such as image classifications, segmentation, localization etc.

CNN like any other artificial neural networks have a input layer, an output layer and various hidden layers. However a CNN works slightly different than a normal neural network. Computers read images as pixels in a matrix form of N*N*3 (height by width by depth) and a CNN analyzes these image pixel values to perform its operations.

A CNN mainly has four important layers which form the central aspects of working of a CNN. Those are-

1. Convolution Layer : The convolutional layer makes use of a set of learnable filters which are used to detect the presence of specific features or patterns present in the original image (input). It is expressed in the form of M*M*3. A filter has a smaller dimension than the original image but the same depth. This filter is convolved (slided) across the width and height of the input image and a dot product is computed from the two matrices, the image and the filter.

2. Activation Layer : Activation function is a node that is put at the end of the layers of neural networks through which it decides to fire or not. The output from this layer is fed to the subsequent layers. The most commonly used activation function in CNN is ReLu. Other activation functions are sigmoid, tanh, maxout etc.

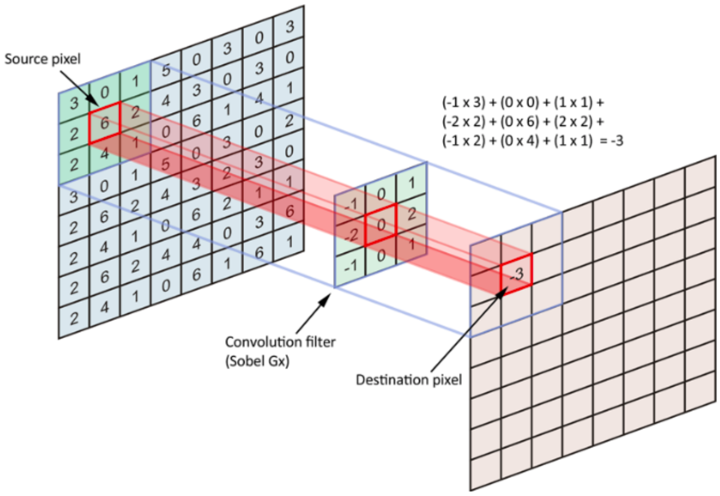

Figure 3: Working of Convolution layer in a CNN

Figure Source: An Intuitive Guide to CNN by Daphne Cornelisse - Medium

3. ReLu : Rectified Linear Unit or more commonly known as ReLu as discussed above is one of the most commonly used activation function in CNNs. It is a piece wise linear function that will output the input directly if is positive, otherwise, it will output zero. It is known as piece wise because is linear for half of the input domain and non-linear for the other half. Advantages of ReLu over other activation functions like sigmoid, hyperbolic tangent function is that the latter activation functions are prone to vanishing gradient problem which makes it difficult for the model to chose the direction to move the parameters to improve the cost function.

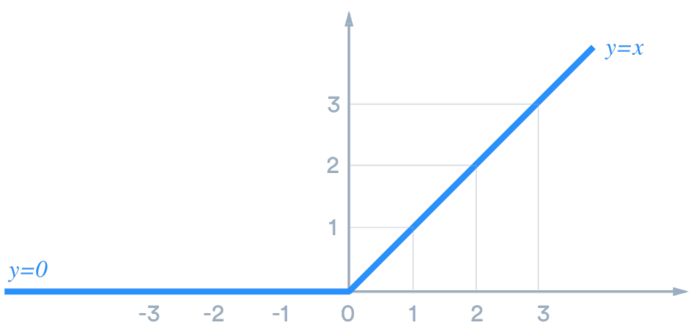

Figure 4: ReLu Activation Function

Figure Source: A gentle introduction to ReLu - machinelearningmastery.com

4. Pooling Layer : The main purpose of a Pooling layer is to down sample or to reduce the parameters in the network. The size of the matrix decreases once it passes through the pooling layer. Max pooling is most commonly used operation in the pooling layer. As the name suggests it will take the maximum pixel value from the pool. Other operations are Average pooling, Min pooling etc.

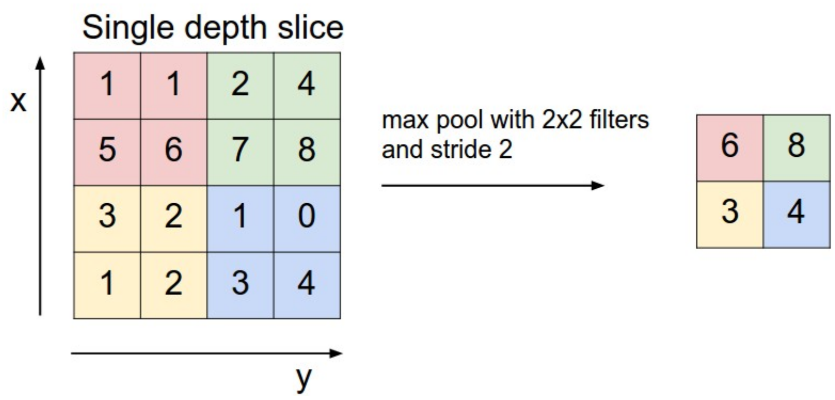

Figure 5: Working of Pooling in a CNN

Figure Source: CS231n by Stanford University

5. Fully Connected Layer : This is the same type of layers found in typical Artificial Neural Networks. It flattens the entire input volume into a single vector. All the nodes are connected to each and every node of the subsequent layers. Generally this layer connects to the final output layer.

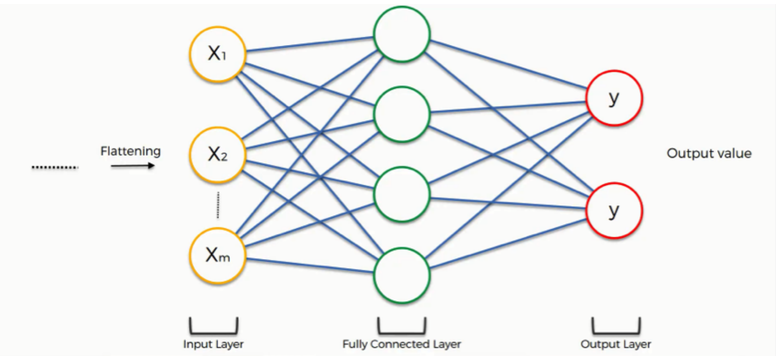

Figure 6: Fully connected layer

Figure Source: superdatascience.com

Other important aspects of a CNN are as follows -

1. Stride: Stride can be defined as the number of pixels by which we move our filter matrix over the input matrix.

2. Kernel size: It defines the size of the filter used in the convolutional layer.

3. Padding: Padding is applied to the input images when the size of the input image is less to the required size. Then padding pads the input volume with zeros around the border. With zero padding the image size will increase but it will not change in the information of the input value.

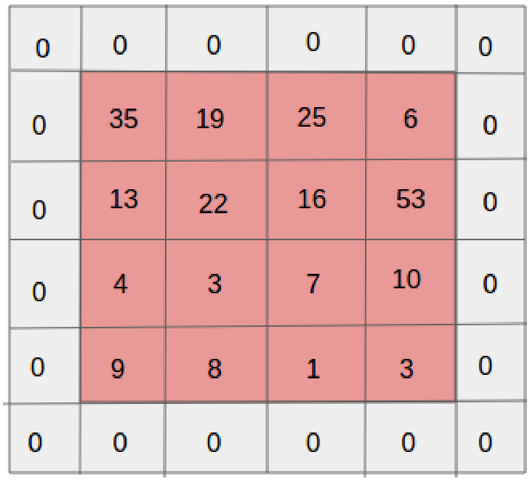

Figure 7: Padding in a CNN

Figure Source: XRDS - ACM MAgazine for students

4. Dropout: Dropout is a regularization technique used for reducing over-fitting in networks. This layer "drops out" a random set of pixel values by setting those values as zero.

5. Batch Normalization: This is a technique used to improve speed, performance and stability of a network by normalizing the output of a previous activation layer by subtracting the batch mean and dividing by the batch standard deviation.

CNN reduces the size of images into a form in which it is easier to process it. CNN consists of a convolutional layer in which is input image is convolved with a filter or kernel of size k. The kernel of size k moves over the image with its stride till it parses one complete image. The objective of the convolutional layer in CNN is to extract features like edges, arch, etc. from the image. Then, the pooling layer in CNN is responsible for reducing the image size to reduce computational intensity to process the data. There are two types of pooling, max pooling and average pooling. Max pooling returns the maximum value from the portion of the image covered with the kernel and average pooling returns the average of all the value covered with the kernel. Classification is done in the fully connected layer, the image is flattened first into a column vector which is fed to the feed forward neural network and backpropagation is applied to every iteration of training. Over a series of epochs, the model learns features of the images and classify them using a classifier.

Chapter 5 Evaluation Metrics

Parameters to evaluate the performance of the network are as follows. Firstly, the classified images can be sub-grouped into four groups to evaluate its performance:

1. True Positives(TP): True positive is an outcome where the classifier correctly predicts the positive class.

2. False Positives(FP): False positive is an outcome where the classifier incorrectly predicts the positive class.

3. True Negatives(TN): True negative is an outcome where the classifier correctly predicts the negative class.

4. False Negatives(FN): False negative is an outcome where the classifier incorrectly predicts the negative class.

		Actual	
		Positive	Negative
Predicted	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

Figure 8: Visualization of True positives, False positives, True negatives and False negatives

Figure Source: Beyond Accuracy - Precision and Recall by Will Koherson - Medium

Now, these sub-groups can be used to calculate a few parameters to evaluate the performance of a particular network:

1. Precision: Precision is defined as the ratio of accurate positives the model claims compared to the total number of positives the model claims.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

2. Recall: Recall is defined as the ratio of the accurate positives the model claims to the total num of positives in the dataset.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

While recall expresses the ability to find all relevant instances in a dataset, precision expresses the proportion of the data points our model says was relevant actually were relevant.

3. F1 Score: F1 Score is the harmonic average of precision and recall.

$$F1Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3}$$

4. Accuracy: Accuracy is defined as the correctly classified images among the total number of images.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

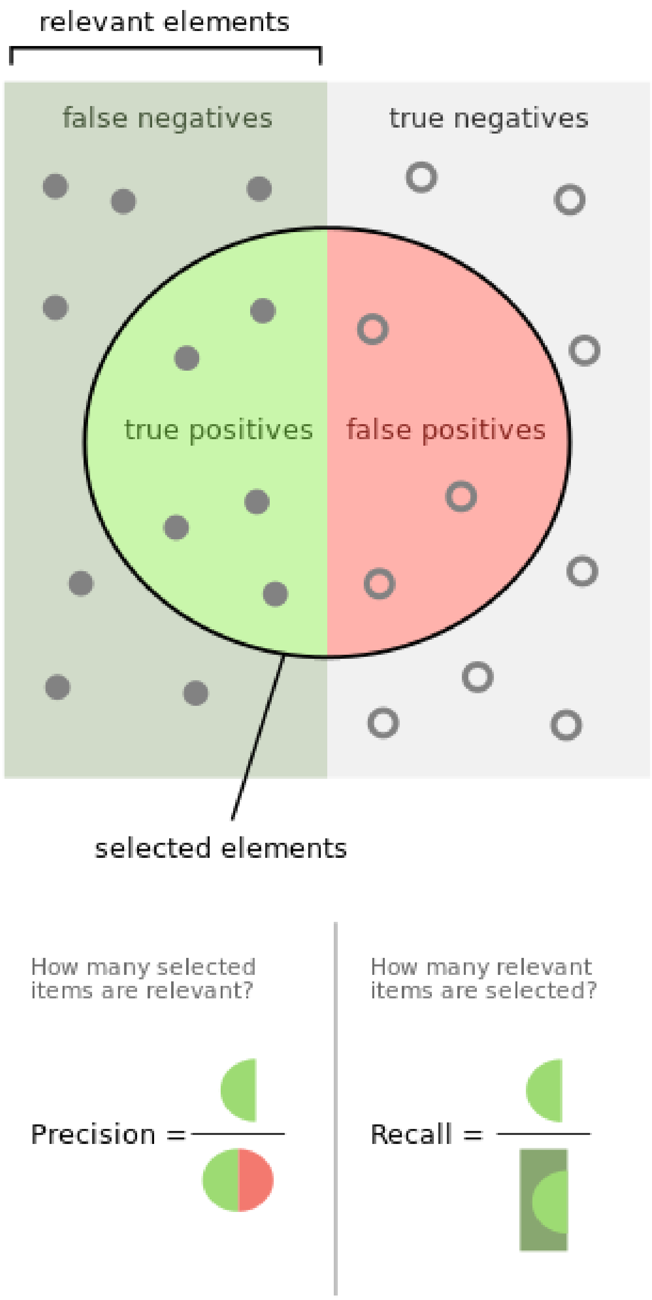

Figure 9: Visualization of Precision and Recall

Figure Source: wikipedia.org

Chapter 6 Architectures

The ten CNN architectures on which we trained the dataset are as follows:

1. LeNet 5
2. AlexNet
3. ZFNet
4. VGG 16
5. VGG 19
6. Inception V3
7. Inception ResNet
8. Xception
9. DenseNet
10. MobileNet

LeNet 5

LeNet 5 was to a great extent utilized for classifying manually written digits on bank cheque books in the United States. LeNet 5 is a convolutional neural network pre-trained on MNIST dataset, i.e. manually written digits from 0 to 9 proposed by Yann LeCun, Leon Bottouand Pattrick Haffner. MNIST has 60,000 training set and 10,000 testing set. LeNet5 is the most straightforward of all state of the art deep learning based computer vision model. LeNet 5 is a significant model since it made character acknowledgment simpler.

Architecturally, LeNet5 is a small network and includes 7 layers, among which 2 are convolutional layers, 2 average pooling layers pursued by flattening convolutional layer and lastly a softmax classifier. Tanh activation function is utilized in all layer in a customary LeNet 5 with the exception of the last layer where softmax classifier is utilized. In our investigation, we have utilized ReLu in each layer and softmax function for classification in the last layer.

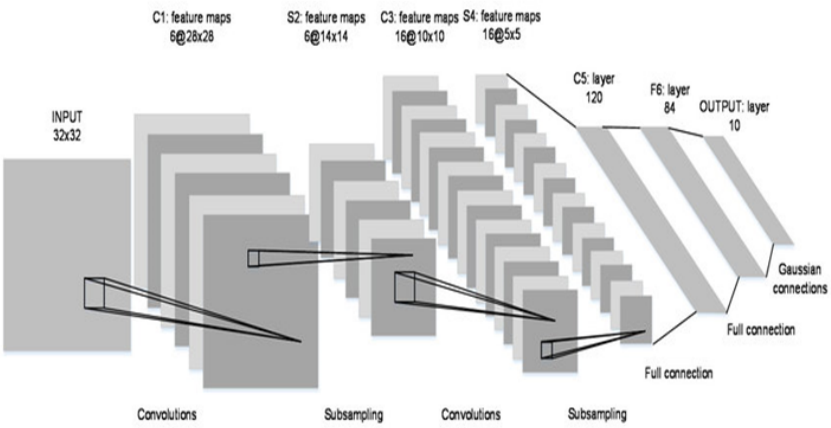

Figure 10: Architecture: LeNet 5

First Layer

Input of LeNet is a 32x32 image which passes the first convolutional layer with 6 feature maps or filters of size 5x5 and stride of 1. The image size of picture changes from 32x32x1 to 28x28x6.

Second Layer

Then the LeNet 5 applies average pooling layer. Average pooling layer has a filter size of 2x2 and stride of 2. The image dimensions reduce to 14x14x6.

Third Layer

Third comprises of 16 feature maps of which just 10 are associated with the 6 feature maps of the previous layer as shown in the figure below. The fundamental motivation to do this was to break the symmetry in the system and keep the quantity of associations in a sensible bound. In the picture below, S2 represents the average pooling layer and C3, the second convolutional layer. This outcomes in an image size of 10x10x16.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

TABLE I

EACH COLUMN INDICATES WHICH FEATURE MAP IN S2 ARE COMBINED BY THE UNITS IN A PARTICULAR FEATURE MAP OF C3.

Figure 11: Second convolutional Layer: LeNet 5

Fourth Layer

Fourth layer is again an average pooling layer with a filter size of 2x2 and a stride of 2. This has 16 feature maps so it lessens the output to 5x5x16.

Fifth Layer

Fifth layer is the fully connected convolutional layer with an feature map of 120 and size 1x1 each. Every one of these 120 nodes are associated with all the 400 nodes in the previous layer.

Sixth Layer

Sixth layer is a fully connected layer with 84 nodes.

Seventh Layer

This is the final output layer of LeNet 5, last layer is fully connected softmax output layer.

ALexNet

AlexNet is a network proposed by Alex Krizhevsky, Ilya Sutskever and advisor Geoffrey Hinton. AlexNet contended in ImageNet Large Scale Visual Recognition Challenge 2012 and accomplished the top error rate of 15.3% versus 26.2% for the second spot. AlexNet is viewed as one of most powerful papers published in Computer Vision. It is pre-trained on ImageNet dataset which has around 12,00,000 training images with 1000 classes.

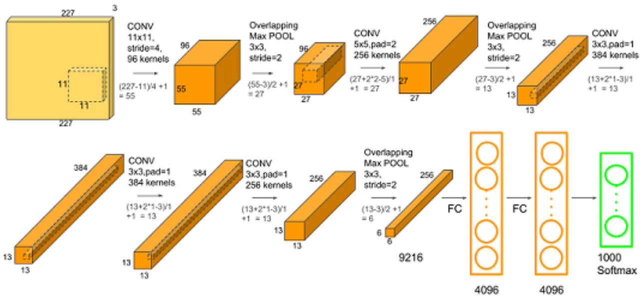

Figure 12: Architecture: AlexNet

Convolutional Layers

AlexNet has 5 convolutional layers, first convolutional layer has a filter size of 11x11 and a stride of 4. Second convolutional layer has a filter size of 5x5 and a stride of 2. Every one of the initial two convolutional layers are followed by overlapping max pooling layers. Further features are extracted using 3 more convolutional layers with a filter size of 3x3 and a stride of 1. These three convolutional layers are connected directly. The fifth convolutional layer is followed by an overlapping max pooling layer. Overlapping max pooling layers has a filter size of 3x3 and a stride of 1. Every one of these layers are followed by ReLU activation function.

Overlapping Max Pooling

Overlapping max pooling layers are usually applied to down sample the height and width of the image. Overlapping max pooling is very similar to max pooling layer, the only difference is the adjacent windows over which max pooling is computed overlap each other.

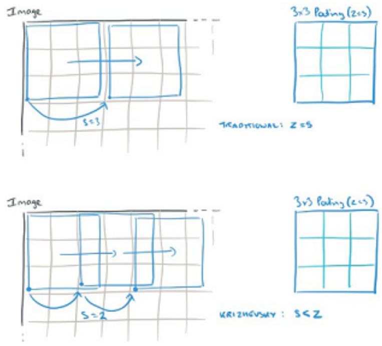

Figure 13: Overlapping Max Pooling Layer: AlexNet

ReLU Non-Linearity

An important feature of AlexNet is its use of Rectified Linear Unit (ReLU) non linearity. The graph below shows that AlexNet could have achieved 25% error rate 6 times faster than an equivalent network using tanh as tested on CIFAR-10 dataset.

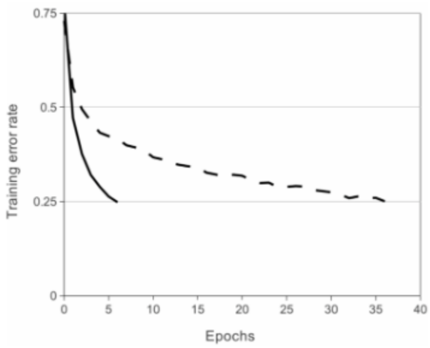

Figure 14: ReLU vs Tanh error rate: AlexNet

Reason for this is as the slope of tanh converges to zero for higher positive and negative values, this slows down the gradient descent for higher positive and negative values. In case of ReLU, the slope does not converge to zero for higher positive values, this makes the optimization to converge faster. For negative value, the slope is zero, but most of the neurons in a neural network end up having a positive value.

Fully Connected Layers

AlexNet has 3 fully connected layers with ReLU as activation function in each of these 3 fully connected layers. Softmax classifier is used for classification.

ZFNet

ZFNet won ImageNet Large Scale Visual Recognition Competition in 2013 by achieving an error rate of 14.8% over 15.2% achieved by AlexNet in 2012. Architecturally, ZFNet is very similar to AlexNet. ZFNet is an optimized version of AlexNet. The bottlenecks of AlexNet were found and removed to achieve a better performance.

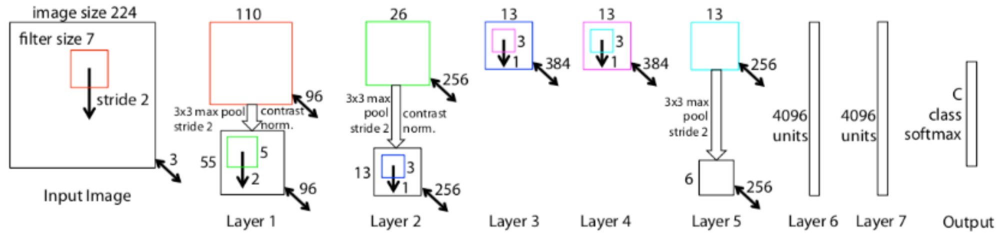

Figure 15: Architecture: ZFNet

In ZFNet, the filter size is reduced from 11x11 to 7x7. This reduced the number of dead features reduced in the first layer. A dead feature is a situation where a convolutional kernel fails to learn any significant representation. Visually it looks monotonic single colour image where all values are close to each other.

In addition to increasing the number of features the number of filters in the network were doubled in the convolutional layers and the number of neurons in the fully connected layers were doubled in the fully connected layers as compared to AlexNet. In AlexNet, there were 48-128-192-192-128-2048-2048 kernels or neurons whereas in ZFNet, there were 96-256-384-384-256-4096-4096 kernels or neurons. This resulted in increase in complexity of the internal representations and hence decrease in error rate.

Apart from these changes, ZFNet is just an evolution of AlexNet with all the layers same.

VGG 16/19

VGG is stand for Visual Geometry Group. The network is 16 and 19 layers deep for VGG16 and VGG19 respectively. VGGNet is a pretrained Convolutional Neural Network (CNN) that has been trained on 1.2 million images from the ImageNet Dataset by the Visual Geometry Group at University of Oxford.

VGGNet is the Keras model of the 16 and 19-layer network used by the VGG team in the the Image Net Large Scale Visual Recognition Challenge (ILSVRC) in 2014 competition. It stood first at the image localization task and stood second on the image classification task.

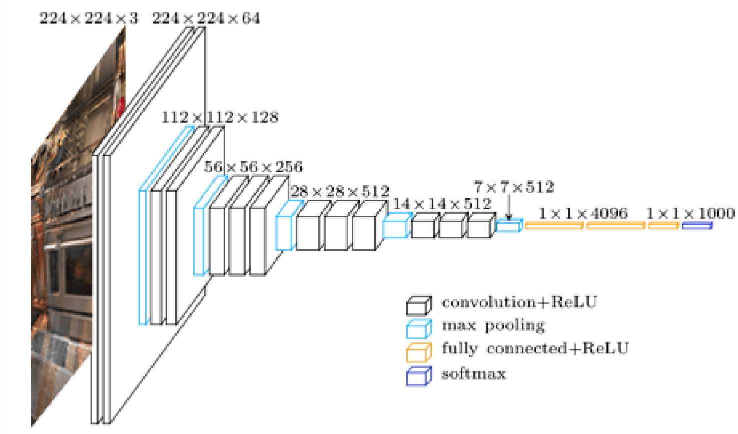

Figure 16: Architecture: VGG16/19

Figure Source:
http://agnesmustar.com/2017/04/19/build-vgg16-scratch-keras-part/

1. First and Second Layers: The input for VGG is a 224*224*3 RGB image which passes through first and second convolutional layers with 64 feature filters having size 3*3 and same pooling with a stride of 14. The image dimensions changes to 224*224*64.Then the VGG16 applies max pooling layer with a filter size 3*3 and a stride of 2. The resulting image dimensions will be reduced to 112*112*64.

2. Third and Fourth Layer: Next, there are two convolutional layer with 128 feature filters having size 3*3 and a stride of 1. Then the VGG16 applies max pooling layer with a filter size of 3*3 and a stride of 2. This layer has 128 feature filters so the output will be reduced to 56*56*128.

3. Fifth and Sixth Layers: These layers are convolutional layers with filter size 3*3 and a stride of 1. Then the VGG16 applies max pooling layer with a filter

size of 3*3 and a stride of 2. This layer has 256 feature filters so the output reduced to 56*56*256.

4. Seventh to Twelfth Layer : These layers are convolutional layers with filter size 3*3 and a stride of 1. Then the VGG16 applies max pooling layer with a filter size of 3*3 and a stride of 2. This layer has 512 feature filters so the output reduced to 7*7*512.

5. Thirteenth convolutional layer output is flatten through a fully connected layer with 25088 feature filters each of size 1*1.

6. Fourteenth and Fifteenth Layers are again two fully connected layers with 4096 units.

7. Output layer has a softmax output layer with 1000 possible values.

Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	224 x 224 x 3	-	-	-
1	2 X Convolution	64	224 x 224 x 64	3x3	1	relu
	Max Pooling	64	112 x 112 x 64	3x3	2	relu
3	2 X Convolution	128	112 x 112 x 128	3x3	1	relu
	Max Pooling	128	56 x 56 x 128	3x3	2	relu
5	2 X Convolution	256	56 x 56 x 256	3x3	1	relu
	Max Pooling	256	28 x 28 x 256	3x3	2	relu
7	3 X Convolution	512	28 x 28 x 512	3x3	1	relu
	Max Pooling	512	14 x 14 x 512	3x3	2	relu
10	3 X Convolution	512	14 x 14 x 512	3x3	1	relu
	Max Pooling	512	7 x 7 x 512	3x3	2	relu
13	FC	-	25088	-	-	relu
14	FC	-	4096	-	-	relu
15	FC	-	4096	-	-	relu
Output	FC	-	1000	-	-	Softmax

Figure 17: Summary of VGG16 Architecture

Figure Source: https://engmrk.com/vgg16-implementation-using-keras/

Inception_v3

Inception_V3, 1st runner up of ILSRV 2015 is a 42-layer convolutional neural network. Inception also uses smaller filter size which again reduces the number of parameters in Inception. In Inception, a filter size of 3x1 followed by 1x3 replaces the traditional 3x3 filters in order to prevent over-fitting. This is known as filter factorization. Additionally, max-pooling layer is applied to the images. Inception_V3 uses RMS_Prop Optimizer, batch normalization in classifiers and label smoothing which gives a reduced error-rate compared to other networks and improved accuracy.

By a carefully crafted design, the width and depth of the network are increased keeping the computational intensity constant. To optimize quality, the architectural decisions were based on the Hebbian principle and the intuition of multi-scale processing.

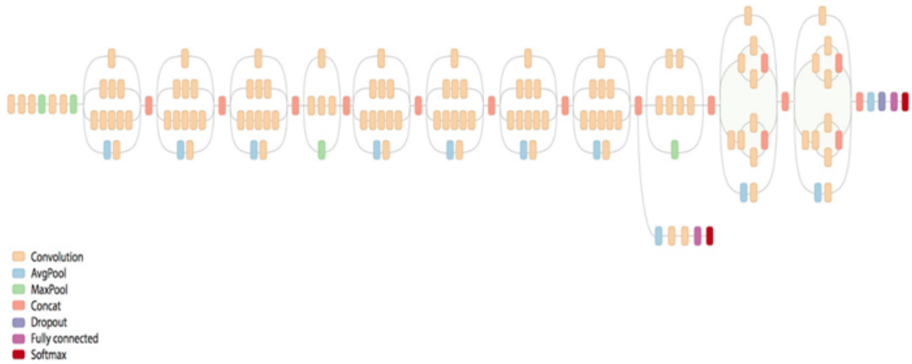

Figure 18: Architecture: Inception_V3

The main highlights of Inception_v3 was that it reduced the computational intensity and increased the accuracy. The premises were as follows:

1. When neural networks do not reduce the dimensions of the input drastically, the network performs better. Reducing the dimensions rapidly leads to loss of information known as representational bottlenecks. Representational bottlenecks are lessened.

2. Smart factorization methods are used to make the computational intensity better.

Factorizing a 5x5 convolution to two 3x3 convolution operations. This improves the speed as a 5x5 convolution is approximately 3 times more computationally intensive than 3x3 convolution. Moreover, a convolution of filter size nxn is factorized using nx1 and 1xn convolutions. The filter banks in the module were made wider. This was done to remove representational bottleneck. These were the few changes made in Inception_V2. Inception_v3 incorporated all these changes and in addition to all these changes, RMS Prop Optimizer, 7x7 factorized convolutions, Batch normalization in auxiliary classifiers and label smoothing. Label soothing prevents the model from over fitting.

Inception_ResNet

Inception_ResNet emerged from Inception_v1 and merged with ResNet which solves the vanishing gradient problem seen in networks. Infact, Inception_v4 and Inception_ResNet were proposed in the same paper. The changes made in Inception_V4 were the initial operations that were done before introducing the Inception blocks. This is known as the stem of the Inception module.

Inception_ResNet is a hybrid model created using ResNet and Inception. In Inception_ResNet residual blocks are introduced that are added to the input of the Inception module.

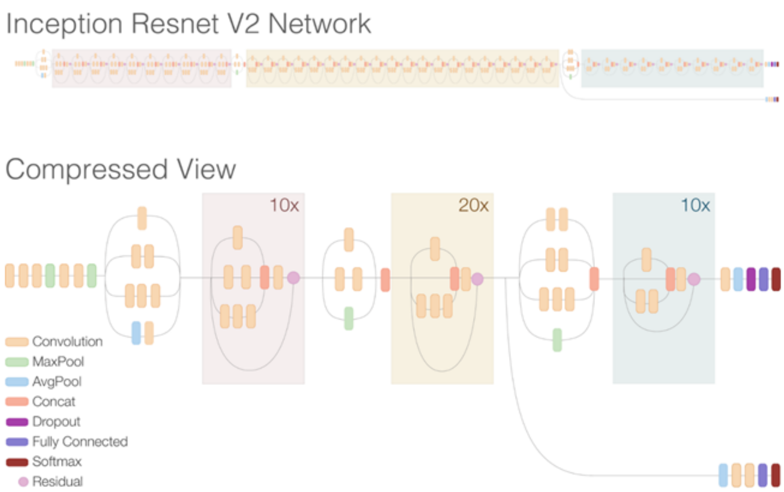

Figure 19: Architecture: Inception_ResNet

To add residuals to the network, both input and output should have similar dimensions. Therefore, here 1x1 convolutions are applied after actual convolution and the depth is increased after convolution to match the size of output image with input image.

The pooling that were done inside the inception blocks were done to to fit the residual connections. Residual units which re deeper tend to die when filter number exceeds 100. Therefore, the residual activations were reduced in the range 0.1 to 0.3.

Xception

Xception, which stands for Extreme version of Inception works on the concept of depthwise separable convolution. The Xception architecture is inspired by inception, and inception modules have been replaced with depthwise separable convolutions. The modified depthwise separable convolution is the pointwise convolution followed by a depthwise convolution.

Xception architecture is trained on ImageNet dataset of 350 million images and 17,000 classes. This architecture has the same number of parameters as inception network. The performance of this architecture increased due to the model parameters not because of increased capacity. - In this architecture there is 36 convolutional layers, which comprise of 14 modules. All of these modules have linear residual connections around them, except for the first and last modules.

The major differences from the inception architecture are given below: The original depthwise separable convolutions as usually implemented perform first channel-wise spatial convolution and then perform 1*1 convolution whereas the modified depthwise separable convolution perform 1*1 convolution first then channel-wise spatial convolution.

There is non-linearity after first operation. In Xception, the modified depthwise separable convolution, there is no intermediate ReLU non-linearity.

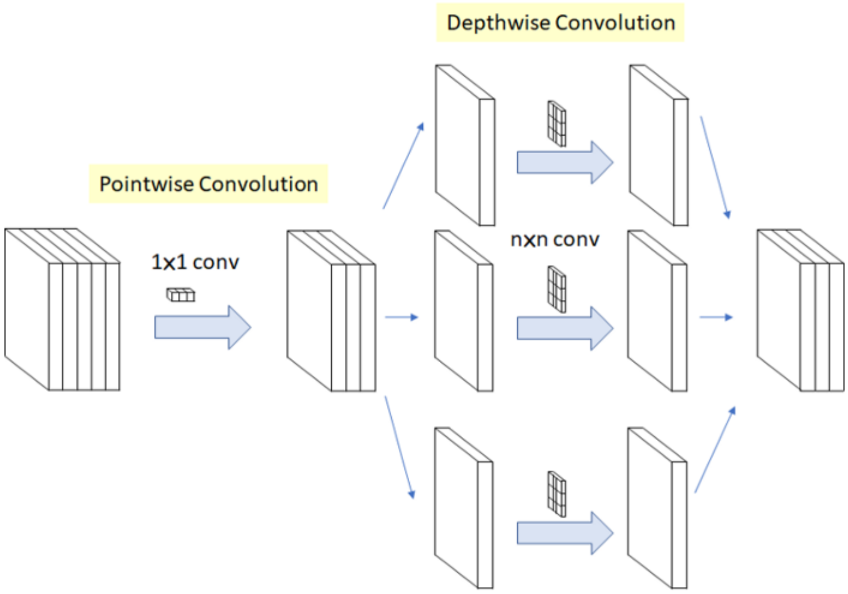

Figure 20: Architecture: Xception

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

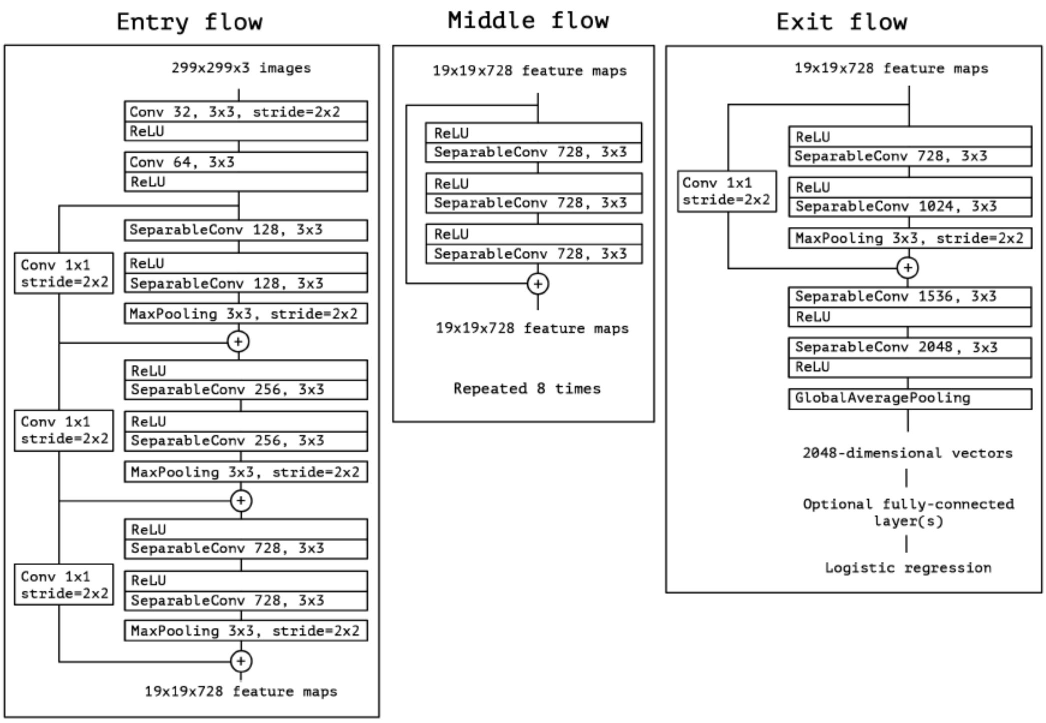

Figure 21: The Xception architecture

Figure source: - Francois Chollet, Google Inc. Xception: Deep Learning with

Depthwise Separable Convolutions - 2017

DenseNet

DenseNet is a logical extension of ResNet. In ResNet to avoid residual errors, a previous layer is merged to future layers using a fundamental block known as identity block whereas in DenseNet outputs from previous layers are concatenated instead of adding. Dense Convolutional Network or DenseNet made an observation that convolutional layers can be denser and more accurate if they contain shorter connection between layers that are closer to the input and output. In DenseNet every layer is connected to every other layer in feed forward fashion. DenseNet with L layers has L(L+1)/2 direct connections. In DenseNet each layer's feature maps of all preceding layers are used as inputs and feature maps of present layers are used as inputs in subsequent layers. DenseNets diminish vanishing gradient problem, strengthen feature propagation, encourage feature reuse and reduce the number of parameters.

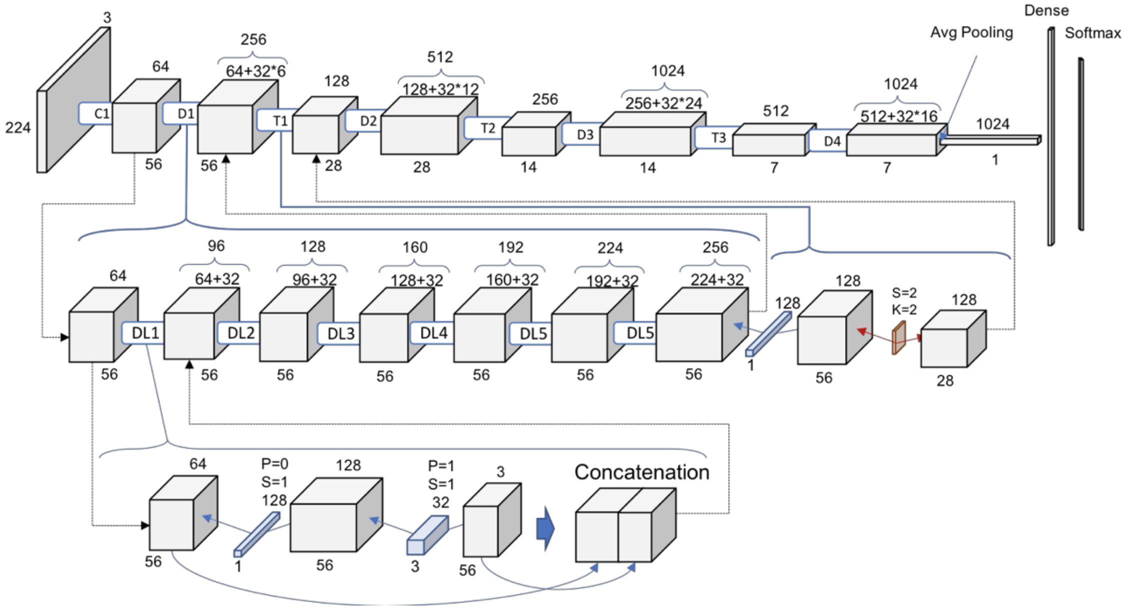

Figure 22: Architecture: DenseNet

In DenseNet, the connectivity patterns require lesser parameters than traditional convolutional neural networks as redundant feature maps are not relearned here. Feature maps are passed on from layer to layer. Each layers learn feature maps from its previous layer and writes to the subsequent layers. In ResNet this is done using identity blocks these layers contribute very less and also may get randomly dropped at times and ResNet become very similar to unrolled like recurrent neural network and also makes ResNets substantially larger because each layer have their weights. DenseNets have very less parameters, they update very less number of feature maps and keep the others same to prevent redundant learning and decision is made depending on feature maps of all the layers. DenseNets are easier to train because of their becuase of its improved flow of information and gradients. Each layer has direct access to the gradients from the loss function and the first information signal, prompting an understood deep supervision. This helps preparing of more profound network architectures. Further, we likewise see that dense connections have a regularizing impact, which lessens over-fitting on errands with smaller training sets.

Concatenating feature maps increases efficiency.Inception networks also concatenate feature maps but DenseNets are simpler and more efficient. In DenseNets, each layer adds k features from its state to the global state. Global state can be accessed from anywhere within the network unlike other traditional networks.

Layers	Output Size	DenseNet-121($k=32$)	DenseNet-169($k=32$)	DenseNet-201($k=32$)	DenseNet-161($k=48$)
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	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 36$
Transition Layer (3)	14×14	1×1 conv			
	7×7	2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$
Classification Layer	1×1	7×7 global average pool			
		1000D fully-connected, softmax			

Figure 23: DenseNet architectures for ImageNet

To improve computational intensity, a 1x1 convolutional layer can be introduced as a bottleneck layer before each 3x3 layer. In DenseNet, each 1x1 convolutional layer reduces to 4k feature maps. These layers are named bottleneck features.

In Dense Net number of feature maps can be reduced using θ also named the compression factor. The value of compression factor varies from 0 to 1. When θ matches 1, the number of feature maps remain the same. When θ is in between 0 and 1, the DenseNet model is referred to as DenseNet-BC.

MobileNet

MobileNet is used in mobile phones and in embedded applications as it is light weight in its architecture. It uses depthwise separable convolution. Depthwise separable convolution performs convolution on each coloured layer rather than combining all three and flattening it.

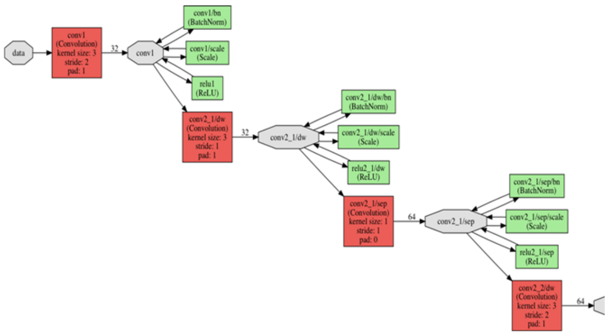

Figure 24: Architecture: MobileNet

For MobileNets the depthwise convolution applies one filter to every input channel. The pointwise convolution then applies a 11 convolution to mix the outputs the depthwise convolution. A regular convolution each filters and combines inputs into a replacement set of outputs in one step. The depthwise divisible convolution splits this into 2 layers, a separate layer for filtering and a separate layer for combining. This factorisation has the impact of drastically reducing computation and model size.

Architecture of MobileNet can be given as follows:

1. convolutional layer (stride 2)

2. depthwise layer
3. pointwise layer, this doubles the number of input channels
4. depthwise layer (stride 2)
5. pointwise layer

MobileNet is a very low maintenance but high performing state of the art convolutional neural network. MobileNet has a multiple accumulates which is a measure of number of fused multiplication and addition, the speed and power consumption depends of the network depends on this.

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	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ 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	$1 \times 1 \times 1000$

Figure 25: Architecture: MobileNet

Chapter 7 Training

We took two approaches to train all the CNN architectures -

1. Patient ID Inclusive - In this we randomly selected the training and validation images from the entire set of patient IDs.

2. Patient ID Exclusive - In this we first randomly selected 75% of the patient IDs and the training set was created exclusively from that set of patient IDs and the validation set was created from the other 25% of the patient IDs. In this case the training and validation set are patient exclusive to each other.

As discussed above, due to limitation of computation power and unbalanced number of images in both the classes, the images were under-sampled to 50,000 images per class for training and 5,000 images per class for validation. By doing this we achieved uniformity between both the classes. The images were selected randomly removing any chance of bias in the dataset. The models were trained on 4 GPU cards of 11 GB RAM each.

We trained the first three CNN architectures (LeNet, AlexNet and ZFNet) from scratch and used transfer learning for training rest of the CNN architectures.

Transfer learning is a machine learning technique where a model is trained on one task and we can use the same model on a different but similar task. The pre trained models were trained on the ImageNet dataset. The pre trained models are available in the Keras library. We used the pre trained weights of the architectures as initial weights for the training, then all the layers of each of the architectures was trained using the Breast Cancer Histopathological Dataset.

As discussed all the CNN architectures were trained entirely with the "ImageNet" weights initialized. We got various level of success for each of the CNN architectures. The accuracy table for each of the CNN architecture with the Patient ID Inclusive approach is as follows-

From this we can see that the first five CNN architectures are not performing

		Precision	Recall	F1 Score
LeNet 5	Non IDC	0	0	0
	IDC	0.5	1	0.67
AlexNet	Non IDC	0.88	0.44	0.58
	IDC	0.63	0.94	0.75
ZFNet	Non IDC	0.89	0.77	0.83
	IDC	0.8	0.91	0.85
VGG16	Non IDC	0.9	0.84	0.87
	IDC	0.85	0.91	0.85
VGG19	Non IDC	0.89	0.86	0.87
	IDC	0.86	0.89	0.88
Inception	Non IDC	0.91	0.84	0.87
	IDC	0.85	0.92	0.88
Inception ResNet v2	Non IDC	0.89	0.83	0.86
	IDC	0.84	0.89	0.87
DenseNet121	Non IDC	0.87	0.88	0.88
	IDC	0.88	0.87	0.88
Xception	Non IDC	0.92	0.85	0.88
	IDC	0.86	0.92	0.0.89
MobileNet v2	Non IDC	0.91	0.86	0.88
	IDC	0.87	0.91	0.89

Table 1: Precision, Recall and F1 Score of Trained Models

that well. Hence for the next parts we did not consider the first five architectures.

For the next approach of Patient ID exclusive approach we trained only the last five CNN architectures namely Inception, InceptionResnet, DenseNet, Xception and MobileNet because they gave better accuracy.
The accuracy table for each of the CNN architecture with the Patient ID Inclusive approach is as follows-

		Precision	Recall	F1 Score
Inception	Non IDC	0.86	0.87	0.86
	IDC	0.87	0.86	0.86
Inception ResNet v2	Non IDC	0.88	0.83	0.86
	IDC	0.84	0.89	0.87
DenseNet121	Non IDC	0.83	0.83	0.83
	IDC	0.83	0.82	0.83
Xception	Non IDC	0.86	0.85	0.86
	IDC	0.85	0.87	0.86
MobileNet v2	Non IDC	0.87	0.86	0.86
	IDC	0.86	0.87	0.86

Table 2: Precision, Recall and F1 Score of Trained Models

From this table we can see that the second approach has performed a little bit poorer that the first approach. This can be attributed to the fact that since all the patient IDs were exclusive to each other the images in the training set and the validation set were very different to each other and more overfitting occured in the second approach than the first approach.

Chapter 8 Ensemble

Ensemble Learning is a machine learning technique which uses multiple machine learning algorithms to obtain better predictive performance than could be achieved from a single algorithm. In it multiple algorithms or learners are trained to solve the same problem. Later all these algorithms or learners are combined in a way to get a better overall performance.

In this case we trained ten CNN architecture on the same dataset and got various level of success from each of the architectures. Finally we took the top five best performing CNN architectures namely Inception, Inception_Resnet, Densenet, Xception and Mobilenet and used all of these architectures to get our final result.

So the basic idea behind using ensemble learning technique is to increase the precision of the model (at the cost of recall). In this case increasing the precision of the model is very important because we want to give as much correct predictions as possible even if we do not predict for some cases where we are not confident enough. This way for the cases which we are not confident we can say ”None”. This way the recall will decrease but the precision will increase dramatically.

We have used five ensemble learning techniques. Those are as follows-

1. Uniform Output
2. Probability Average
3. Probability Average with threshold
4. Majority Voting
5. Majority Voting with threshold

Uniform Output

According to this rule if all the five architectures give the same result for a given image then we accept that result otherwise we assign it 'None'. This way we are increasing the confidence of our model because if we are getting a result then all the five CNN architectures agree to that and are giving the same result.
The final accuracy table incorporating the ensemble technique for the with Patient ID inclusive approach is as follows-

	Precision	Recall	F1 Score
Non IDC	0.97	0.69	0.80
IDC	0.94	0.79	0.86
None	0	0	0

Table 3: Uniform Output - Patient IDs Inclusive

The final accuracy table incorporating the ensemble technique for the with Patient ID exclusive approach is as follows-

	Precision	Recall	F1 Score
Non IDC	0.96	0.70	0.81
IDC	0.94	0.68	0.79
None	0	0	0

Table 4: Uniform Output - Patient IDs Exclusive

Probability Average

According to this rule we take the probability of each of the CNN architectures and average them. If the resulting probability is equal to or more than 0.5 we predict it as IDC otherwise we predict it as Non-IDC.
The final accuracy table incorporating the ensemble technique for the with Patient ID inclusive approach is as follows-

	Precision	Recall	F1 Score
Non IDC	0.92	0.88	0.90
IDC	0.88	0.93	0.90

Table 5: Probability Average - Patient IDs Inclusive

The final accuracy table incorporating the ensemble technique for the with Patient ID exclusive approach is as follows-

	Precision	Recall	F1 Score
Non IDC	0.89	0.87	0.88
IDC	0.87	0.90	0.88

Table 6: Probability Average - Patient IDs Exclusive

Probability Average with threshold

According to this rule we take the probability of each of the CNN architectures and average them. After that instead of using the usual 0.5 as the threshold we can set our own threshold. For example in this case we set the following threshold. For the image to be classified as IDC the probability should be greater than 0.9 i.e 80% probability that the image is IDC and for the image to be classified as Non-IDC the probability should be less than 0.1 i.e 20% probability.
By doing this we achieved very high precision, infact highest precision among all the ensemble models but also achieved the lowest recall.
The final accuracy table incorporating the ensemble technique for the with Patient ID inclusive approach is as follows-

	Precision	Recall	F1 Score
Non IDC	0.99	0.51	0.67
IDC	0.97	0.60	0.74
None	0	0	0

Table 7: Probability Average with threshold - Patient IDs Inclusive

The final accuracy table incorporating the ensemble technique for the with Patient ID exclusive approach is as follows-

	Precision	Recall	F1 Score
Non IDC	0.97	0.63	0.76
IDC	0.96	0.57	0.72
None	0	0	0

Table 8: Probability Average with threshold - Patient IDs Exclusive

Majority Voting

According to this rule if three or more CNN architectures give the same output then we predict that output.
The final accuracy table incorporating the ensemble technique for the with Patient ID inclusive approach is as follows-

	Precision	Recall	F1 Score
Non IDC	0.92	0.88	0.90
IDC	0.89	0.92	0.90

Table 9: Majority Voting - Patient IDs Inclusive

The final accuracy table incorporating the ensemble technique for the with Patient ID exclusive approach is as follows-

	Precision	Recall	F1 Score
Non IDC	0.89	0.87	0.88
IDC	0.87	0.89	0.88

Table 10: Majority Voting - Patient IDs Exclusive

Majority Voting with threshold

According to this rule atleast four architectures have to give the same output for a given image only then we predict that output otherwise we predict ”None”.

The final accuracy table incorporating the ensemble technique for the with Patient ID inclusive approach is as follows-

	Precision	Recall	F1 Score
Non IDC	0.95	0.83	0.88
IDC	0.91	0.88	0.89
None	0	0	0

Table 11: Majority Voting with threshold - Patient IDs Inclusive

The final accuracy table incorporating the ensemble technique for the with Patient ID exclusive approach is as follows-

	Precision	Recall	F1 Score
Non IDC	0.93	0.81	0.86
IDC	0.90	0.83	0.86
None	0	0	0

Table 12: Majority Voting with threshold - Patient IDs Exclusive

Conclusion

From the previous section we can see that by using the ensemble techniques we can achieve higher precision for both Non IDC and IDC more than any of the CNN architectures individually. Hence, combining the results of the five best performing neural networks we achieve higher precision.

Future Scope

This project can we used as a preliminary step to Invasive Ductal Carcinoma (IDC) detection in patients under the supervision of doctors and pathology experts. Future Works can include deploying this project in a open source server where remote hospitals can upload the biopsy images and can get predictions without installing a state-of-the-art facility. If implemented, this can help in early IDC detection and classification of millions of people who might lack the resources to avail the facilities. This will not only save money and resources but also give fast predictions which would help the doctors in quick responses.

References

[1] cancer.org

[2] ncbi.nlm.nih.gov

[3] cancerindia.org

[4] kaggle.com/paultimothymooney/breast-histopathology-images

[5] medicalnewstoday.com

[6] cancercenter.com

[7] hopkinsmedicine.org/breast_center

[8] medium.com

[9] machinelearningmastery.com

[10] iosrjournals.org

[11] pyimagesearch.com

[12] cs231n.github.io

[13] adeshpande3.github.io

[14] arxiv.org

[15] cnnblogs.com

[16] fightcancer.org

[17] shodhganga.inflibnet.ac.in

[18] wikipedia.org/wiki/Computer-aided_diagnoses

[19] thesis.bilknet.edu

[20] sciencedirect.com

[21] oreilly.com/library/view/deep-learning

[22] blogs.nvidia.com

[23] image-net.org

[24] journals.uic.edu

[25] academia.edu

[26] towardsdatascience.com

[27] image-net.org

[28] mathsworks.com

[29] springer.com

[30] analyticsvidya.com

[31] analyticsindiamag.com

[32] learnopencv.com

[33] discovery.ucl.ac.uk

[34] datasciencecentral.com

[35] cs231n.stanford.edu

[36] ieee.xplore.ieee.org

[37] Breast Cancer Histopathological Image Classification using Convolutional Neural Networks-2016 Fabio A Spanhol et al.

[38] Mitko Veta, Josien P. W. Pluim, Paul J. van Diest, and Max A. Viergever, Breast Cancer Histopathology Image Analysis: A Review - 2014

[39] Majid Nawaz, Adel A. Sewissy, Taysir Hassan A. Soliman, Multi-Class Breast Cancer Classification using Deep Learning Convolutional Neural Network - 2018

[40]Daniel Lvy, Arzav Jain, Breast Mass Classification from Mammograms using Deep Convolutional Neural Networks - 2016

[41] Teresa Arau jo, Guilherme Aresta, Eduardo Castro1, Jose Rouco, Paulo Aguiar, Catarina Eloy, Anto nio Polo nia, Aure lio Campilho, Classification of breast cancer histology images using Convolutional Neural Networks - 2017

[42] Hiba Chougrad, Hamid Zouaki, Omar Alheyane, Deep Convolutional Neural Networks for Breast Cancer Screening - 2018

[43] Sertan Kaymak, Abdulkader Helwan, Dilber Uzun , Breast cancer image classification using artificial neural networks - 2017

[44] Saira Charan, Muhammad Jaleed Khan, Khurram Khurshid, Breast Cancer Detection in Mammograms using Convolutional Neural Network 2018

[45] Abdullah-Al Nahid Mohamad Ali Mehrabi, and Yinan Kong, Histopathological Breast Cancer Image Classification by Deep Neural Network Techniques Guided by Local Clustering - 2018