

# Breast Cancer Classification Using Neural Networks

Rohan Raman

Vellore Institute of Technology

**Abstract-** Breast cancer (BC) is the most common type of cancer in adult women. The need for a system for early detection of BC is evident and consequently there is a demand for it in the healthcare industry. This manuscript attempts to solve this problem by proposing a solution based on convolution neural networks. The model is trained on histopathological images as opposed to the normally used mammography images for better accuracy from the BreakHis dataset. Various pre-existing architectures were tried, ultimately a trimmed down version of the VGG-16 architecture gave the best results. The proposed model was evaluated using various metrics and gave better results than most of the pre-existing models.

**Keywords-** Convolution Neural Networks, Neural Networks, Breast Cancer, Deep learning, Histopathology

## I. INTRODUCTION

The National Cancer Institute defines histopathology as "the study of diseased cells and tissues using a microscope". The target tissue is obtained from the body and then stained with a suitable colouring agent to make the features visible easily. This stained tissue is then placed on a slide and observed under a microscope for further study. A histopathology report is also called a biopsy report.

This manuscript attempts to classify the images into 5 categories based on the features from the histopathological images. Benign being non-cancerous while malignant, that is cancerous being further classified as ductal carcinoma, lobular carcinoma, mucinous carcinoma and papillary carcinoma.

1. Benign



2. Ductal Carcinoma



3. Lobular Carcinoma



4. Mucinous Carcinoma

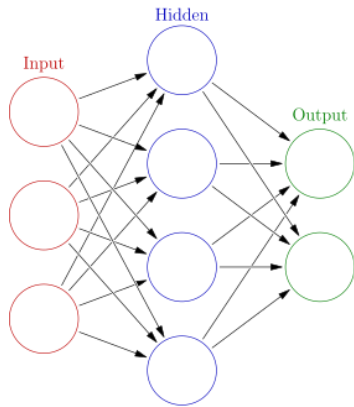


5. Papillary Carcinoma



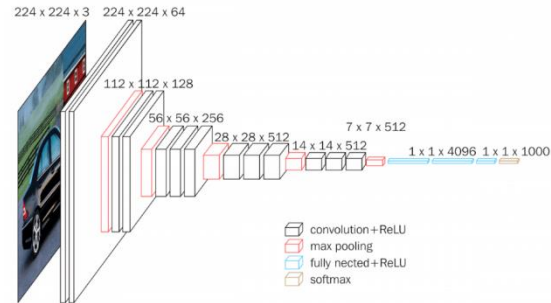
A neural network is based on how the human brain works. Artificial neuron is conceived as an analogy of biological neuron, where it takes a number of signals as input, and fires an output if the combined input signal strength exceeds a certain threshold. When a number of artificial neurons are connected together, it is called an artificial neural network. The ability of an artificial neuron is limited if it simply performs a weighted sum of all inputs, so we pass the weighted sum of inputs to a non-linear function called the activation function. The hidden layer in neural networks generally refers to the biases to be added to the weighted sum along with the activation function to which it is passed. The hidden layer is expected to extrapolate or highlight the salient features from the input layer and give the result as an output layer. Neural networks given enough data can find patterns in almost anything and is thus used with unstructured data like video, text, images, speech etc. A basic neural network is shown in the diagram below.

Deep learning is a set of techniques that utilize the concept of a neural network. Here I have used convolution neural networks to extract the features from the histopathological images and train the model on these features.



- 13 Convolution Layers
- 3 Fully Connected Layers
- Activation function: Rectified Linear Unit (ReLU)

The block diagram is given below:



VGG16 has about 138 million trainable parameters, as elaborated in the following table:

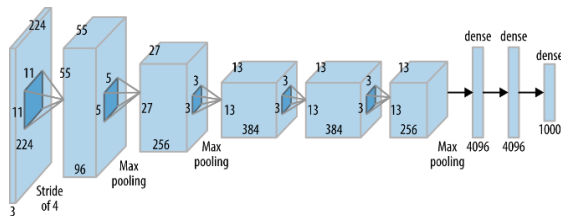
#	Input Image	output	Layer	Stride	Kernel	in	out	Param
1	224 224 3	224 224 64	conv3-64	1	3 3	3	64	1792
2	224 224 64	224 224 64	conv3-64	1	3 3	64	64	36928
	224 224 64	112 112 64	maxpool	2	2 2	64	64	0
3	112 112 64	112 112 128	conv3-128	1	3 3	64	128	73856
4	112 112 128	112 112 128	conv3-128	1	3 3	128	128	147584
	112 112 128	56 56 128	maxpool	2	2 2	128	128	65664
5	56 56 128	56 56 256	conv3-256	1	3 3	128	256	295168
6	56 56 256	56 56 256	conv3-256	1	3 3	256	256	590080
7	56 56 256	56 56 256	conv3-256	1	3 3	256	256	590080
	56 56 256	28 28 256	maxpool	2	2 2	256	256	0
8	28 28 256	28 28 512	conv3-512	1	3 3	256	512	1180160
9	28 28 512	28 28 512	conv3-512	1	3 3	512	512	2359808
10	28 28 512	28 28 512	conv3-512	1	3 3	512	512	2359808
	28 28 512	14 14 512	maxpool	2	2 2	512	512	0
11	14 14 512	14 14 512	conv3-512	1	3 3	512	512	2359808
12	14 14 512	14 14 512	conv3-512	1	3 3	512	512	2359808
13	14 14 512	14 14 512	conv3-512	1	3 3	512	512	2359808
	14 14 512	7 7 512	maxpool	2	2 2	512	512	0
14	1 1 25088	1 1 4096	fc			1	25088	4096
15	1 1 4096	1 1 4096	fc			1	4096	16781312
16	1 1 4096	1 1 1000	fc			1	4096	4097000
Total								138,423,208

There are mainly 4 model architectures that are considered for image classification:

**Alexnet:**

- 5 Convolution Layers
- 3 Fully Connected Layers
- Activation function: Rectified Linear Unit (ReLU)

The block diagram is given below:



Alexnet has about 62 million parameters, as elaborated in the following table:

Input	Output	Layer	Stride	Pad	Kernel size	in	out	# of Param
227 227 3	55 55 96	conv1	4	0	11 11	3	96	34944
55 55 96	27 27 96	maxpool1	2	0	3 3	96	96	0
27 27 96	27 27 256	conv2	1	2	5 5	96	256	614656
27 27 256	13 13 256	maxpool2	2	0	3 3	256	256	0
13 13 256	13 13 384	conv3	1	1	3 3	256	384	885120
13 13 384	13 13 384	conv4	1	1	3 3	384	384	1327488
13 13 384	13 13 256	conv5	1	1	3 3	384	256	884992
13 13 256	6 6 256	maxpool5	2	0	3 3	256	256	0
		fc6			1 1	9216	4096	37752832
		fc7			1 1	4096	4096	16781312
		fc8			1 1	4096	1000	4097000
Total								62,378,344

**VGG16:**

**ResNet18:**

- 17 Convolution Layers
- 1 Fully Connected Layers
- Activation function: Rectified Linear Unit (ReLU)

In order to tackle the vanishing gradient problem ResNet introduced two types of shortcut connections, the identity shortcut and the projection shortcut.

ResNet18, eighteen being the number of layers in the network has about 11 million trainable parameters.

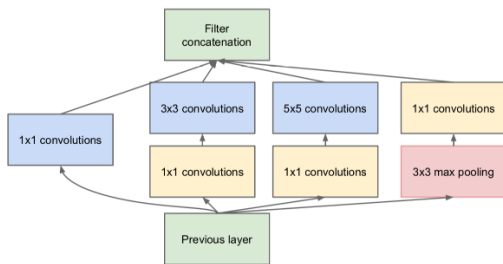
ResNet18 - Structural Details										
#	Input Image	output	Layer	Stride	Pad	Kernel	in	out	Param	
1	227 227 3	112 112 64	conv1	2	1	7 7	3	64	9472	
2	112 112 64	56 56 64	maxpool	2	0.5	3 3	64	64	0	
3	56 56 64	56 56 64	conv2-1	1	1	3 3	64	64	36928	
4	56 56 64	56 56 64	conv2-2	1	1	3 3	64	64	36928	
5	56 56 64	56 56 64	conv2-3	1	1	3 3	64	64	36928	
6	56 56 64	28 28 128	conv3-1	2	0.5	3 3	64	128	73856	
7	28 28 128	28 28 128	conv3-2	1	1	3 3	128	128	147584	
8	28 28 128	28 28 128	conv3-3	1	1	3 3	128	128	147584	
9	28 28 128	28 28 128	conv3-4	1	1	3 3	128	128	147584	
10	28 28 128	14 14 256	conv4-1	2	0.5	3 3	128	256	295168	
11	14 14 256	14 14 256	conv4-2	1	1	3 3	256	256	590080	
12	14 14 256	14 14 256	conv4-3	1	1	3 3	256	256	590080	
13	14 14 256	14 14 256	conv4-4	1	1	3 3	256	256	590080	
14	14 14 256	7 7 512	conv5-1	2	0.5	3 3	256	512	1180160	
15	7 7 512	7 7 512	conv5-2	1	1	3 3	512	512	2359808	
16	7 7 512	7 7 512	conv5-3	1	1	3 3	512	512	2359808	
17	7 7 512	7 7 512	conv5-4	1	1	3 3	512	512	2359808	
18	1 1 512	1 1 1000	fc	7	0	7 7	512	1000	513000	
Total										11,511,784

ResNet has a number of variations, ResNet50 and ResNet101 being the most commonly used.

**Inception:**

Each inception network consists of a several modules. Every inception module essentially comprises of four operations in parallel

- 1x1 conv layer
- 3x3 conv layer
- 5x5 conv layer
- max pooling



The respective outputs from the four parallel operations are concatenated in a depth wise fashion in the filter concatenation block. The structural details of GoogleNet, a type of inception net, are given below:

GoogleNet - Structural Details										
Input Image	output	Layer	Input Layer	Stride	Pad	Kernel	in	out	Param	
227 227 3	112 112 64	conv1	input	2	1	7 7	3	64	9472	
112 112 64	56 56 64	maxpool	conv1	2	0.5	3 3	64	64	0	
56 56 64	56 56 64	conv2-1	conv2-1	1	1	3 3	64	64	36928	
56 56 64	56 56 64	conv2-2	conv2-2	1	1	3 3	64	64	36928	
56 56 64	56 56 64	conv2-3	conv2-3	1	1	3 3	64	64	36928	
56 56 64	28 28 128	conv3-1	conv3-1	2	0.5	3 3	64	128	73856	
28 28 128	28 28 128	conv3-2	conv3-2	1	1	3 3	128	128	147584	
28 28 128	28 28 128	conv3-3	conv3-3	1	1	3 3	128	128	147584	
28 28 128	28 28 128	conv3-4	conv3-4	1	1	3 3	128	128	147584	
28 28 128	14 14 256	conv4-1	conv4-1	2	0.5	3 3	128	256	295168	
14 14 256	14 14 256	conv4-2	conv4-2	1	1	3 3	256	256	590080	
14 14 256	14 14 256	conv4-3	conv4-3	1	1	3 3	256	256	590080	
14 14 256	14 14 256	conv4-4	conv4-4	1	1	3 3	256	256	590080	
14 14 256	7 7 512	conv5-1	conv5-1	2	0.5	3 3	256	512	1180160	
7 7 512	7 7 512	conv5-2	conv5-2	1	1	3 3	512	512	2359808	
7 7 512	7 7 512	conv5-3	conv5-3	1	1	3 3	512	512	2359808	
7 7 512	7 7 512	conv5-4	conv5-4	1	1	3 3	512	512	2359808	
1 1 512	1 1 1000	fc	conv5-4	7	0	7 7	512	1000	513000	
Total										11,511,784

Initially I was looking at Resnet, VGG and inception but I realised that with our dataset defeats the purpose of using inception net as a consistent portion of every image consists of the tumor. Resnet wasn't used because the skip connection was only considerably useful in very deep neural networks, whereas if we go that deep with the proposed model, the model will most likely overfit. So, I decided to use VGG and applied necessary changes by removing certain layers to improve the accuracy as it was more suited to our use case.

**III. METHODOLOGY**

**Data Preparation:**

The model was trained on the histopathological images from the BreakHis dataset. The dataset consists of over 8000 images categorised as malignant and benign. It was further split up into the various types of malignant and benign tumors. The subcategories contained images in various magnifications.

First, the images of all the magnifications were grouped and placed together under the name of the subcategory. For the benign images, the subcategory was

removed and all the images were labelled as benign. This give us our 5 classes for classification, as ductal carcinoma, lobular carcinoma, mucinous carcinoma papillary carcinoma and benign.

**Image Augmentation:**

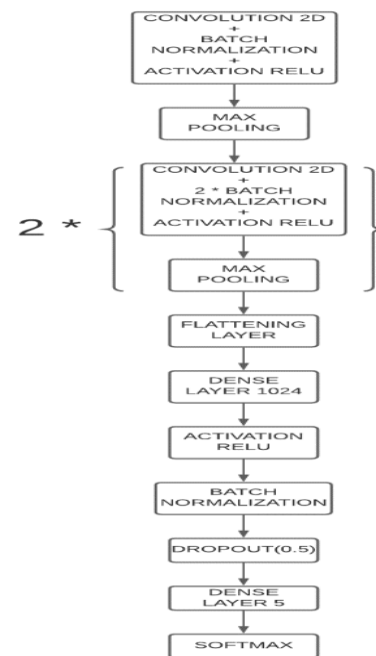
Deep learning models require a considerable amount of training data to produce desirable results. Image augmentation is a way of supplementing the original dataset with processed images in the form of random rotations, shifts etc thereby boosting the performance of the model. For image augmentation I used the ImageDataGenerator function in the keras library. The code for augmentation is hosted at <https://github.com/romanronin/breast-cancer-histopathology-classification> under the name 'augmentation.py'.

**Model:**

About the model:

1. Removed dropout layers so that the number of parameters is not reduced, to get a better accuracy.
2. Increased the filters for the convolution layers (32-64-128) respectively.
3. Reduced the pool size for better and finer patterns.
4. First dense fully connected layer has 1024 dimensions.
5. The second dense fully connected layer has 5 dimensions for classification into five classes.
6. Batch size is 55.
7. The learning rate is set to  $1 \times 10^{-3}$ .

The detailed architecture:



The model summary is given below:

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 96, 96, 32)	896
activation_8 (Activation)	(None, 96, 96, 32)	0
batch_normalization_6 (Batch Normalization)	(None, 96, 96, 32)	128
max_pooling2d_4 (MaxPooling2D)	(None, 32, 32, 32)	0
conv2d_7 (Conv2D)	(None, 32, 32, 64)	18496
activation_9 (Activation)	(None, 32, 32, 64)	0
batch_normalization_7 (Batch Normalization)	(None, 32, 32, 64)	256
conv2d_8 (Conv2D)	(None, 32, 32, 64)	36928
activation_10 (Activation)	(None, 32, 32, 64)	0
batch_normalization_8 (Batch Normalization)	(None, 32, 32, 64)	256
max_pooling2d_5 (MaxPooling2D)	(None, 16, 16, 64)	0
conv2d_9 (Conv2D)	(None, 16, 16, 128)	73856
activation_11 (Activation)	(None, 16, 16, 128)	0
batch_normalization_9 (Batch Normalization)	(None, 16, 16, 128)	512
conv2d_10 (Conv2D)	(None, 16, 16, 128)	147584
activation_12 (Activation)	(None, 16, 16, 128)	0
batch_normalization_10 (Batch Normalization)	(None, 16, 16, 128)	512
max_pooling2d_6 (MaxPooling2D)	(None, 8, 8, 128)	0
flatten_2 (Flatten)	(None, 8192)	0
dense_3 (Dense)	(None, 1024)	8389632
activation_13 (Activation)	(None, 1024)	0
dense_4 (Dense)	(None, 5)	5125
activation_14 (Activation)	(None, 5)	0
Total params: 8,674,181		
Trainable params: 8,673,349		
Non-trainable params: 832		

## IV. RESULTS

The overall accuracy of the proposed system is 90.45 %. Other evaluation metrics were also used like the confusion matrix, precision, recall and f1 score. The metrics are summarized below.

```

[[829  3  8 20  4]
 [ 33 354 12  4 15]
 [  3 15 231  1  6]
 [ 17  4 29 268  3]
 [ 20  2  1  5 260]]
testing accuracy= 0.9045179319981369

```

	precision	recall	f1-score	support
Benign	0.96	0.92	0.94	902
Ductal_Carcinoma	0.85	0.94	0.89	378
Lobular_Carcinoma	0.90	0.82	0.86	281
Mucinous_Carcinoma	0.83	0.90	0.87	298
Papillary_Carcinoma	0.90	0.90	0.90	288
accuracy			0.90	2147
macro avg	0.89	0.90	0.89	2147
weighted avg	0.91	0.90	0.90	2147

### Confusion matrix:

It is a  $N * N$  matrix used for evaluation of classification problems. Where  $N$  is the number of classes, hence the  $5 * 5$  matrix. It gives a holistic view of our model and the errors in it. The rows represent the predicted values and the columns represent the actual values. Ideally, the main diagonal should have values much larger as compared to the rest of the matrix as is the case in our matrix.

### Precision:

Precision by definition is the true positives divided by the total predicted positives. It gives us an idea out of the predicted positives how many are actually positive. The higher the better.

### Recall:

Recall by definition is the true positives divided by the total actual positives. It gives us an idea out the actual positives how many were correctly identified as positive. The higher the better.

### F1 score:

F1 score is the harmonic mean between precision and recall. It takes into account both the false positives and false negatives. It is commonly used as a metric for model selection besides accuracy. The higher the better.

## REFERENCES

- [1] Aqeel Anwar (2019, June 7) Difference between AlexNet, VGGNet, ResNet, and Inception. *Towards Data Science*. From <https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaecccc96>
- [2] Ragab DA, Sharkas M, Marshall S, Ren J. 2019. Breast cancer detection using deep convolutional neural networks and support vector machines. *PeerJ* 7: e6201 <https://doi.org/10.7717/peerj.6201>
- [3] Suki Lau (2017, July11) Image Augmentation for Deep Learning. *Towards Data Science*. From <https://towardsdatascience.com/image-augmentation-for-deep-learning-histogram-equalization-a71387f609b2>