Detection of Invasive Ductal Carcinoma in Pathological Images

Zeyu Fu
Boston University
zeyufu@bu.edu

Yuanrong Liu
Boston University
yliu@bu.edu

Yu Liu
Boston University
liuyu1@bu.edu

Abstract

An increasing need of detecting breast cancer (BCa) regions in whole slide images (WSI) encourages studies on efficient automatic detection. In this work, we consider the detection of Invasive Ductal Carcinoma (IDC), which is a common sub-type of breast cancer, using three Convolutional Neural Network (CNN) models: AlexNet, DenseNet and VGG. Our methods are evaluated on a WSI dataset which consists of 162 whole mount images scanned specimen images from 162 patients diagnosed with IDC. From that, 277,524 patches of size 40x images are exacted. The AlexNet we built reaches an accuracy of 85% which is the score achieved by a previous paper published on Medical Imaging. Further, we employ a more complex model, DenseNet, and a simpler model, VGG. Our methods yield the best quantitative results for automatic detection of IDC regions in terms of regular accuracy and balanced accuracy (90.90%, 90.50%). By comparing the results produced by three different models, we prove that, when a large number of samples are available for training, deep learning approaches are capable at scanning and finally classifying malignancy regions of tissue images for diagnosis support.

1 Introduction

Nowadays, cancers have become the leading cause of human death around the world, millions of people suffer various cancer diseases. The breast cancer related death has one of the highest death rates for woman, and approximately 252,710 breast cancer new cases are estimated to be diagnosed in 2019 in the U.S.[1]. Invasive Ductal Carcinoma (IDC) is the most common sub-type of breast cancers (nearly 80% breast cancers), it refers to cancer that has crossed through the wall of the milk duct and begun to invade the tissues of the breast. However, previous studies have suggested that early detection and suitable treatment for breast cancer could reduce the death rates dramatically[2]. This disease could be commonly identified by pathologists through visual analysis of hematoxylin and eosin (H&E) stained patient tissue slides, and the assessment of disease aggressiveness grading is highly related to the region that was influenced by cancer[3]. The first step for the pathological diagnosis is usually determining whether the region is healthy tissue or not. However, this step is difficult and challenging for pathologists, because the majority of the tissues are benign rather than malignancy[4]. Therefore, accurately diagnosis for the features from the patient tissues could be an important clinical task, and automated methods can be used to save time and reduce error.

In previous studies, one common and effective method for the feature extractions from the medical images is segmentation of histological primitives[5], and utilizes these features for classification methods. However, this method is not always effective. Because this pre-processed step involves handcrafted features by manual selection, it is complicated due to the large size and the complex structure for histological images, and the final classification result is dependent on this proceeding step. Manual steps might lose information for data, and ignore some significant features for the classification, and these bias will reduce the accuracy of the classification. Recently, many features learning methods don't involve these manual steps. They are more straightforward approaches that

they are fully dependent on the data, and don't need any histological knowledge and avoid the handcrafted feature.

One of these methods is the Convolutional Neural Network (CNN). CNN is currently state-of-art generation of tool using for the pattern recognition and decision making missions in digital pathology[6]. In the domain of histological and medical imaging, CNNs have been proved effective for solving many classification and regression problems[7]. For instance, diabetic retinopathy screening, bone disease prediction and age assessment, and etc. CNNs could firstly implement multiple non-linear transformation methods for the original data, and then yield abstract and useful features presentation in the unseen space. With sufficient size of the annotated training data, these hidden features could be finally recognized by the convolutional kernel, and neuron inside the CNNs structure, and the fitted CNN models could be utilized as a predictor for the unannotated testing data. Thus, no manual selection will be generated in this approach.

In this project, we use Convolutional Neural Network on an Invasive Ductal Carcinoma images dataset from Kaggle[8]. All 277,524 patches of 50 $\times$ 50 pixels images were sampled from 162 breast cancer patients whole slide images. We will implement 3 different network structures, including AlexNet, VGG, and DenseNet, and compare their result with former research paper result[9].

The remaining part of the report is organized as follows: Section II will briefly introduce some basic concepts of CNN and some other related works. Section III will describe the dataset and our pre-processing strategies. Section IV will cover the architecture of our CNN models. Section V, and VI will present our final results and discussions.

2 Related Work

2.1 Convolutional Neural Network

Convolutional Neural Network is a class of Neural Network, it is usually applied to analyze visual images. This method implements feature detectors which can filter among whole images, and then measure the correspondence between groups of pixels and patterns of features. Common CNN includes convolutional layers, pooling layers, dense layers, and several methods are used to regularize the model. Most common CNN could be several combinations of convolutional layers (CONV, including activation function) and pooling layers (POOL). fully connected layers (FC) is the last part of CNN, and follow behind all CONV and POOL.

Convolutional Layers: The CONV is the core part of the CNN, and it performs most of the heavily computational stuff. Each layer is a filter of the whole input image, it is a square kernel sized k $\times$ k, and smaller than the size of whole images. All these filters are initialized randomly. Similar to the normal neural network, they will be considered as parameters during optimization, and learned by the network continuously.

During each computation step of one CONV, the dot product will be extracted as one point in the result or so-called feature map. So the original input images will be smaller after going through the filter (Figure 1). This step could extract some high-level features for the image, they could be colors, edges, and etc.

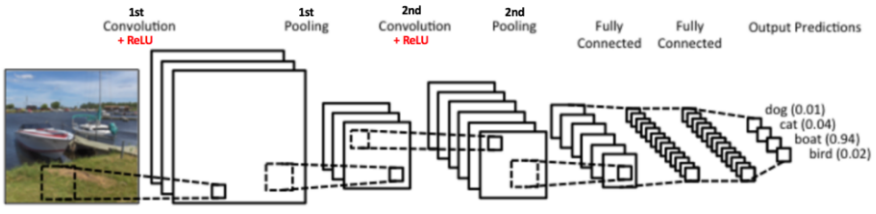

Figure 1: A visualization of the basic CNN configuration[10].

In addition to the basic CONV, activation functions are also implemented during the computation. Recently, the most popular activation function is the ReLU function which represented the rectified

linear unit. Activation functions could convert the linear computation in the layer to fit the non-linear features and patterns in the images. Therefore, CONV could solve the problem in our daily life.

Pooling Layers: POOL often follows behind the CONV. The purpose of this layer is to reduce the size (Width $\times$ Height) of the output in the previous CONV, or make the image smaller for the next CONV. The down-sampling mechanism in POOL is beneficial for the network, because it could reduce the number of the parameters and against the over-fitting. The smaller size of the image is also easier to manage, and the filter of the POOL could suppress the noise from the previous CONV. Common options for the POOL is max POOL or mean POOL.

Dense Layers: Dense Layers are also called fully connected layers (LC). It is usually the last stage in the CNN model. FC is the traditional multiple layer perceptron structure. The purpose of FC is to classify the data. As we discussed before, the CONV and POOL could generate high-level features in the network. Thus, the FC could utilize these input from the previous layers, and classify them with the high-level feature into various classes. Classification functions are also recruited in this step, common functions are the Sigmoid or the Softmax function.

Other methods: In addition to all these layers described above, batch normalization and dropout layers are also implemented in some of our network configurations. Batch normalization could scale the image to increase the training speed, and have some regularization effect. The dropout layer could close some neurons in the previous layers to cut the connection for some neurons in the two neighbor layers. Thus, it could help to reduce the over-fitting problem in some degrees.

2.2 Previous Research Results

Our dataset came from two original publication[4][11]. These two publications used classical machine learning algorithms and the Alexnet to solve this binary classification problem. The results for the previous papers are listed below (Table.1). The result shows that the Convolutional Neural Network performed better than these traditional images analyzing methods that implemented handcrafted features, and get 84.23% balanced accuracy.

Methods	Balanced Accuracy
Convolutional Neural Network	0.8423
Gray Histogram	0.7337
Fuzzy Color Histogram	0.7874
HSV Color Histogram	0.6022
RGB Histogram	0.7724
JPEG Coefficient Histogram	0.7126
Local Binary Partition Histogram	0.6048
MPEG7 Edge Histogram	0.6979
Haralick features	0.6199
Graph-based features	0.6009

Table 1: The accuracy of previous publications

3 Dataset

In the previous work of Cruz-Roa et al.[4], each WSI is segmented into non-overlapping patches of 50×50 pixels via grid sampling. Further, the patches will be discarded if they only contain the slides backgrounds or fatty tissues. A patch is annotated as positive if more than 80% of its region falls into the positive region marked by a pathologist. Otherwise, the image will be annotated as negative. As Figure 2 illustrates, a WSI is annotated by a pathologist and then the WSI will be segmented by grid sampling. The green blocks are positive samples and the left are negative.

As one can interpret, the red grids outnumber green ones. Therefore, it is necessary to balance the dataset before starting the training process.

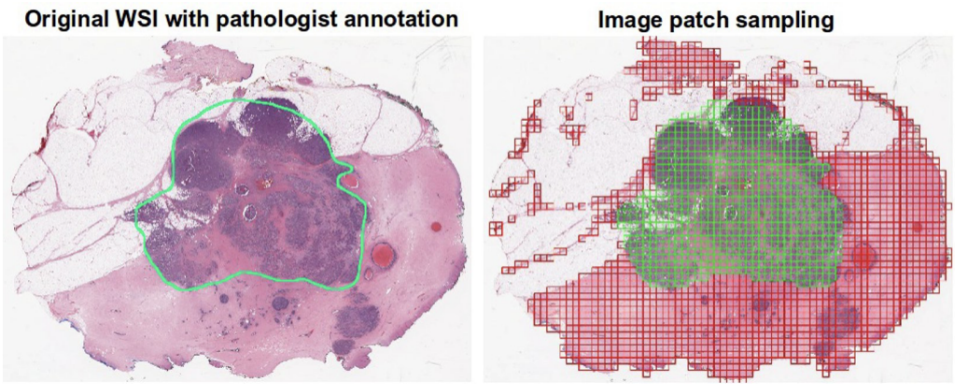

Figure 2: Grid sampling on a whole slide image

Data Balancing: From 162 WSI, 277,524 patches of size 50 $\times$ 50 are sampled out, 198,738 of those samples are IDC negative and the other 78,786 images are IDC positive. To process the imbalanced dataset, we choose to double the size of the positive examples by randomly flip them horizontally, vertically or both. After the implementations are done, the new ratio of positive and negative samples comes to 1:1.26, which is more reasonable.

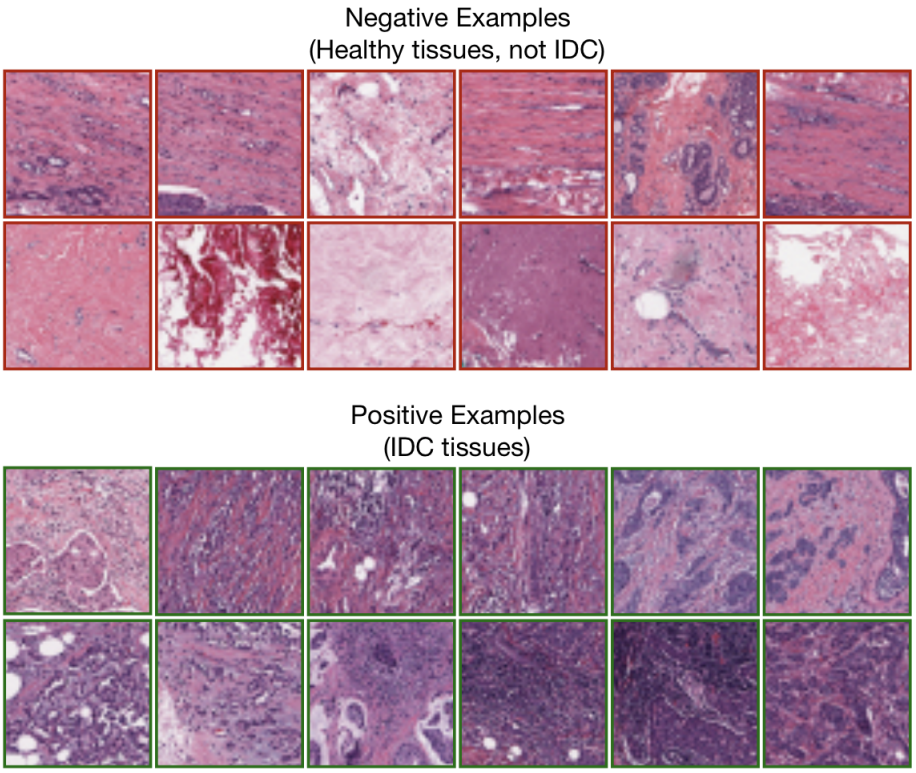

Figure 3: Samples of input images.

Filtering: Nevertheless, not all of the patches from the dataset are well processed. Some of the patches only consist the margin of the WSI it is extracted from. Apart from that, either the height or width of an image sample can be less than 50 pixels. In order to keep the consistency of the size of input images, we need to filter out the images with different sizes.

Image Patches Separation: After the dataset of 162 histopathology slides is manually delineated by expert pathologists, it is separated into three sections: training, validation and testing. About 20% of the total images are assigned to the test set for the sake of final evaluation. Among the rest 80% of the dataset, 70% of which belongs to the training set. The rest are all subjected to the validation set for the purpose of parameter exploration. I,e, 199,534 images in the training set, 85,514 images in the validation set and 71,262 in the testing set. Some examples of the input images are shown in figure 3 for appreciation the difference between images insides the same category (positive versus negative).

Performance Evaluation: By calculating true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN), we can evaluate the performance of IDC detection. In addition to the regular accuracy, we also calculate Balanced Accuracy (BAC) defined in equation (1) where sensitivity is the true positive rate and specificity is the true negative rate. By obtaining BAC, we are able to asses the trade-off occurred when we attempt to minimize the FT and FN rate simultaneously.

$$BAC = \frac{Sensitivity + Specificity}{2} \tag{1}$$

4 Experiments

There are various CNN models in use, we used a simple structure network at first, then implemented other complicated network configurations. AlexNet, DenseNet, VGG were three models we chose to solve the binary classification problem. All experiments recruited the Keras sequential function, and utilized the tensorflow backend. Our code and training process https://github.com/liandan542/IDC-Classification

4.1 AlexNet

According to the previous publications[4], we firstly decided to use a similar CNN configuration with the paper. This structure is a simple version of the Alexnet[12], and the BVLC Alexnet in the Caffe. The detailed structure is listed below (Figure 4). Plus, there is one Batch Normalization layer after each convolutional layer and one dropout layer before each full connected layer in this configuration. ReLu activation function is used for all convolutional layers and fully connected layers.

ConvNet Configuration									
Our Alexnet	Conv5	Maxpool	Conv5	Meanpool	Conv5	Meanpool	FC	FC	Softmax

Figure 4: Configuration of our Alexnet model.

The paper used Adadelta as optimizer, but it only got 82% accuracy by setting different learning rates in our experiment. Finally, we selected the RMSprop method as the optimizer, and we set 1e-6 as the learning rate. Although it was lower than common cases, higher learning rates caused the fluctuation during the increase of validation accuracy and loss. After 100 epochs, the balanced accuracy was 83.10%. And the accuracy could reach up to 84%, which is really similar to the publication result, by running more epochs.

4.2 DenseNet

In order to get higher accuracy on the testing set, we tried another CNN architecture which is DenseNet. Densely Connected Convolutional Networks[13] is published in 2017 and get a test error rate of only 4.2% on CIFAR-10 dataset.

An important and frequently used structure in DenseNet is a dense block. In a single dense block, a layer is connected to all its subsequent layer, which is illustrated in Figure 5. This turns out that DenseNet has L(L+1)/2 direct connections between L layers, while traditional CNNs only have L connections. Built in this way, DenseNet is considerably big and has a huge number of model parameters. The complete architectures of different versions of DenseNet are defined as follows.

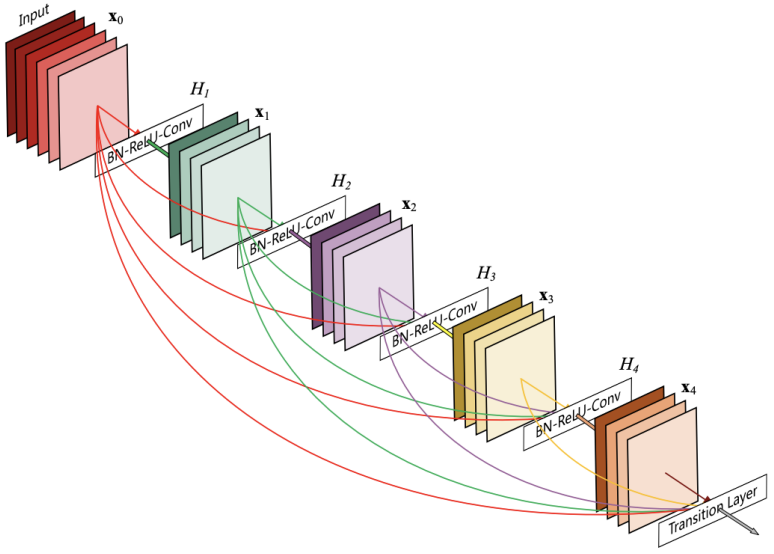

Figure 5: Configuration of our AlexNet model.

Utilizing a pre-defined DenseNet121 model, and training with full-load training data, we found that there is an obvious fluctuation on the curves of validation accuracy and loss. This is probably caused by several reasons: (1) the batch size of training is too small; (2) the learning rate is too large so that the fit does not converge; (3) the parameters of model are of big amount so that the optimization has a high degree of freedom; (4) the data is not good enough and very noisy. To solve this problem, we tried some adjustment we can do on the training process: to increase the batch size, to decrease the learning rate, and to simplify the model architecture.

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	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

Figure 6: Configuration of our AlexNet model.

While increment on batch size is not working well, decreasing learning rate to solve the problem of fluctuation. The only defect is that the accuracy is still not improving. Considering the size of our training images (which is 50 × 50), we assume that we do not need so many model parameters to fit our data. We deleted some dense blocks and left only three dense blocks in the network. By doing this, it turns out we get a better test accuracy of 85.28% and balanced accuracy of 85.39%.

4.3 VGG

Due to the limitation of small input image size, we constructed a simpler VGG network instead of adopting the model from Keras. This VGG model, meanwhile, is the simplest model we have assessed. As the configuration chart illustrates below, a standard VGG16 model has 16 weight layers[14], but we reduced to 10 layers. Thus, the training time is decreased as well.

ConvNet Configuration													
VGG 16	16 weight layers	Conv3 Conv3	Max Pool	Conv3 Conv3	Max Pool	Conv3 Conv3 Conv3	Max Pool	Conv3 Conv3 Conv3	Max Pool	Conv3 Conv3 Conv3	Max Pool	FC	Soft Max
Our VGG	10 weight layers	Conv3 Conv3 Conv3	Max Pool	Conv3 Conv3 Conv3	Max Pool	Conv3 Conv3 Conv3	Max Pool	FC	Soft Max				

Figure 7: Configurations of the VGG models.

In the beginning, we choose Adadelta as optimizer. However, under different learning rate, the best accuracy we can achieve is about 85%, which is not good enough. Finally, with Adam as the optimizer and a learning rate of 1e-4, we get the best accuracy of 90.50% in terms of balanced accuracy.

5 Results and Discussion

In this project, we are using three CNN architectures including AlexNet, DenseNet and VGG. The summary of accuracy is in Table 2.

Model Architecture	AlexNet	DenseNet	VGG
Regular ACC	0.8337	0.8528	0.9090
BAC ACC	0.8310	0.8539	0.9050

Table 2: Accuracy of different models

Comparing to the accuracy of some existing methods (84.23% BAC[4] and 83.68% BAC[11]), our model gained a significant improvement. This is mainly owed to more selections on model architectures, and also our training strategy. The methods described in reference papers are mainly using AlexNet, which is one of the early versions of CNN architecture. In this project, however, we looked through more CNN architectures and selected more recent CNN architecture with better performances.

Our adjustments on model parameters and training parameters also help us improve the performances. One main problem in our training process is fluctuation, this is caused by the conflict between the comparable small size of our training images (image patches) and a comparable large number of our model parameters.

For future work, one extremely meaningful work is to not only detect IDC in small image patches, but also detect IDC region in whole scale slide images. This work will need a lot of data, which is a huge process.

6 Conclusion

We present a binary classifier for Invasive Ductal Carcinoma detection in breast cancer slide images. The model we trained has the best performance of 90.50% balance accuracy on over 70,000 testing examples, which is very convincing. This work will help with automatic detection on breast cancer, which is more efficient than clinical diagnosis.

References

[1] “U.S. Breast Cancer Statistics.” *Breastcancer.org* , www.breastcancer.org/symptoms/understand_bc/statistics.

[2] Migowski, Arn. "Early detection of breast cancer and the interpretation of results of survival studies/A deteccao precoce do cancer de mama e a interpretacao dos resultados de estudos de sobrevida." Ciencia saude coletiva 20.4 (2015): 1309-1310.

[3] Elston, Christopher W., and Ian O. Ellis. "Pathological prognostic factors in breast cancer. I. The value of histological grade in breast cancer: experience from a large study with long-term follow-up. CW Elston IO Ellis. Histopathology 1991; 19; 403–410: AUTHOR COMMENTARY." Histopathology 41.3a (2002): 151-151.

[4] Cruz-Roa, Angel, et al. "Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks." Medical Imaging 2014: Digital Pathology. Vol. 9041. International Society for Optics and Photonics, 2014.

[5] Petushi, Sokol, et al. "Large-scale computations on histology images reveal grade-differentiating parameters for breast cancer." BMC medical imaging 6.1 (2006): 14.

[6] Gertych, Arkadiusz, et al. "Convolutional neural networks can accurately distinguish four histologic growth patterns of lung adenocarcinoma in digital slides." Scientific reports 9.1 (2019): 1483.

[7] Rakhlin, Alexander, et al. "Deep convolutional neural networks for breast cancer histology image analysis." International Conference Image Analysis and Recognition. Springer, Cham, 2018.

[8]Breast Histopathology Images – 198,738 IDC(-) image patches; 78,786 IDC(+) image patches; Paul Mooney; https://www.kaggle.com/paultimothymooney/breast-histopathology-images

[9] Janowczyk, Andrew, and Anant Madabhushi. "Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases." Journal of pathology informatics 7 (2016).

[10] An Intuitive Explanation Of Convolutional Neural Networks Ujjwalkarn - https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

[11] Bower, J. M., and D. Beeman. *The Book of Genesis: Exploring Realistic Neural Models with the General Neural Simulation System.* Springer, 1998.

[12] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[13] Huang, Gao, et al. "Densely connected convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[14] Hassan, Muneeb ul. “VGG16 - Convolutional Network for Classification and Detection.”, 21 Nov. 2018, neurohive.io/en/popular-networks/vgg16/.