Invasive Ductal Carcinoma Breast Cancer Prediction Using Convolutional Neural Network

Vincent Ciptadi
Department of Computer Science
School of Computer Science
Bina Nusantara University
West Jakarta, Indonesia
vincent.ciptadi@binus.ac.id

Gabrielle Angelica Ivandi
Department of Computer Science
School of Computer Science
Bina Nusantara University
Central Jakarta, Indonesia
gabrielle.ivandi@binus.ac.id

Angelia
Department of Computer Science
School of Computer Science
Bina Nusantara University
West Jakarta, Indonesia
angelia007@binus.ac.id

***Abstract*— Breast cancer is the second most severe cancer of all known cancers. If it is not detected quickly, the risk of death from breast cancer can increase exponentially, especially because of the very rapid population growth. Computer-aided diagnosis systems showed potential for improving diagnostic accuracy. In this study, we use a convolutional neural network method for breast cancer image classification.**

Keywords*— *cancer, breast cancer prediction, artificial neural network, convolutional neural network, deep learning.

I. INTRODUCTION

Breast cancer disease is a malignant growth in breast fleshy tissue. Invasive ductal carcinoma (IDC) is the most common phenotypic subtype of all Breast cancers. Breast malignancy can be identified by an abnormality in breasts, an adjustment fit as a fiddle, dimples on the skin, areola-producing liquid, changes in the areola in the recent past, or a red or covered fix on the skin. Among dangerous causes of breast cancer progression in females are obesity, lack of physical exercise, etc. [1]. Results of breast cancer growth are contingent on malignancy type, degree of sickness, and individual age [2].

Breast cancer is one of the most hazardous types of cancer among women in the world. The world health organization's International Agency for Research on Cancer (IARC) estimates that more than 400,000 women expire each year with breast cancer [3]. In 2019, approximately 268,600 new cases of invasive breast cancer and 48,100 cases of DCIS will be diagnosed among US women, and 41,760 women will die from this disease [4]. The growth of breast cancer in Indonesia accounts for 30.5% of all cancers diagnosed and 21.5% of cancer-related deaths among females [5].

Mammography is the gold standard for breast cancer screening as it is the most cost-effective imaging modality; However, one of the most reliable breast cancer detection methods is the examination of histopathological images under a microscope by an expert pathologist. Histopathological images are obtained by examining the biopsy or resection of the specimen under light microscopy of glass slides in the pathology laboratory [6]. With the advent of digital image scanners, the identification of images has become more effective, and it is possible to diagnose with a digital workflow instead of traditional diagnostic techniques.

Advances in computer technology have brought deep neural network learning to the forefront of undertaking complex clinical challenges. Convolutional Neural Network (CNN) has made great strides in medical imaging analysis. CNN algorithms are trained in a manner that allows automatic extraction of features from the input feed that are crucial to the defined problem domain. This process improves its ability to study the input features in an end-to-end manner, using complex, stacked layers to predict the desired output [7]. Therefore, CNN feature extraction is not a variable with each new histopathology image and thus results are consistent.

The purpose of this study is to develop Invasive Ductal Carcinoma breast cancer prediction using a Convolutional Neural Network using breast histopathology images as our dataset.

II. LITERATURE REVIEW

In the following, some studies conducted in this domain are discussed.

Ebrahim and Zhi Feng [3] study the early detection of abnormalities in the breast for diagnosing breast cancer easily. The experiment was carried out using the Wisconsin Diagnosis Breast Cancer database to classify breast cancer as either benign or malignant, consists of 400 observations of patients with breast cancer among which 300 are benign and 100 are malignant status. Supervised learning algorithm - Support Vector Machine (SVM) with kernels like Linear, and Neural Network (NN) are used for comparison to achieve this task. The performances of the models are analyzed where the Neural Network approach provides more 'accuracy' and 'precision' as compared to the Support Vector Machine technique in the classification of breast cancer detection and seems to be a fast and efficient method.

Sangapu Venkata et al. [1] apply deep learning techniques in their study by using RNN (Recurrent Neural Network) to predict the formation of breast cancer disease so that doctors will perform the diagnosis more properly. The used breast cancer dataset was belonging to the UC Irvine repository consisting of 561 instances and 31 attributes. The dataset is being preprocessed by the standardization of the data. The model design comprises model training and testing. The RNN architecture was designed with 3 hidden layers and the total nodes in each hidden layer were increased twice consecutively. Precision, recall, accuracy, and the f1 score of the proposed method are used as the performance evaluation. The results showed good scores and the proposed technique performed well. Based on experimental results, the RNN model exhibited a 97% of f1 score.

Richard Ha et al. [7] proposed a method for predicting the molecular subtype of breast cancer in 216 patients. They used

a dataset in the form of breast photos based on the MRI (Magnetic Resonance Imaging) feature. The proposed algorithm is to use the CNN (Convolutional Neural Network) algorithm. A CNN architecture was designed with 14 layers. Parameters were tuned based on a 20% validation group. A class balanced holdout set of 40 patients was utilized as the testing set. Software code was written in Python using the TensorFlow module on a Linux workstation with one NVidia Titan X GPU. The testing set accuracy was measured at 70%. Therefore, MRI analysis of breast cancer using CNN can predict the molecular subtypes of breast cancer, whereas a larger data set would make it possible to improve the model.

Mingyu Chen et al. [8] proposed convolutional neural networks to assist pathologists in the classification and detection of gene mutation in liver cancer. The histopathological H&E images from the Genomic Data Commons Databases, including 402 WSIs (Whole-Slide Images) of HCC (Hepatocellular Carcinoma) and 89 WSIs of normal liver tissue were used to train a neural network (inception V3) for automatic classification. The model design was divided into 85% training set and 15% testing set with ratio 3/4 of the dataset, and internal-external validation set with ratio 1/4 of the dataset. The performance level was close to the ability of a 5-year experience pathologist, with 96.0% accuracy for benign and malignant classification, and 89.6% accuracy for well, moderate, and poor tumor differentiation.

III. PROPOSED ALGORITHM

The proposed system is based on a designing method for detecting breast cancer at the early stages. Convolutional Neural Networks (CNN) are used to processing the matching and preparing the training dataset. We train our neural networks on the training data and then evaluate performance on the test data using classification metrics: accuracy, precision, recall, f1 score, and specificity. This system is designed based on 2 classes, infected and uninfected. This training outcome model is used to diagnose the test image whether it is infected or not.

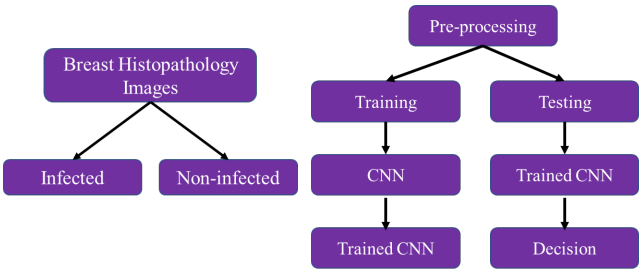

Figure (1). The main proposed system

In figure (1), we can see that the proposed method consists of three main stages. The first stage is to do data pre-processing. The image will be resized and will be mapped based on the label. Next, the image dataset will be split into train dataset and test dataset with proportion 80% for training and 20% for testing. The under-sampling method will also be used to balance the total of feature image dataset according to the labels.

After the pre-processing of the data is completed, the training process for the defined Convolutional Neural Network (CNN) model will be executed using the train dataset. After the model has been trained, the testing process will then be carried out using the trained model with a test dataset. The model will predict the given dataset and provide a classification whether the data belongs to the infected or uninfected class.

The CNN architecture that we use in this study has 8 layers, which consist of the convolutional layer, pooling layer, and flattening layer, dense layer, and dropout layer.

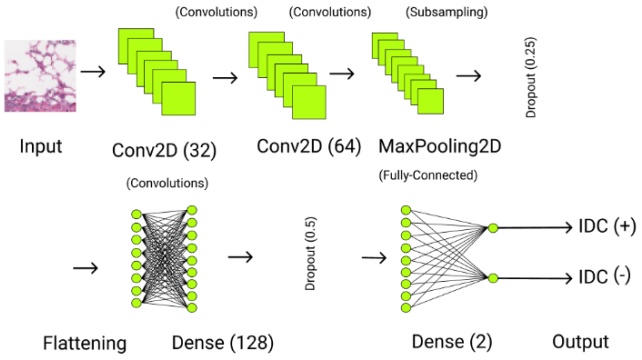

Figure (2). The proposed model architecture

The first and second layer is useful for performing convolutional operations on training images with Conv2D. Here is the formula for Convolutional:

$$z^l = h^{l-1} * W^l$$

Where:

1. z^l is the pre-activation output of layer l
2. h^l is the activation of layer l
3. $*$ is the discrete convolution operator
4. W is learnable parameters.

For the first layer, Conv2D layer is added, and the number of neurons was set into 32. The kernel dimensions are assigned into 3×3. This layer using ReLU as the activation function. The input shape is (50, 50, 3) which is belong to the size of the feature image and the pixel intensity that represents the RGB type. Then, the last parameter is used to determine how much the filter shifts, here the strides value is 2, which means the filter will shift by 2 pixels.

In the second layer, the Conv2D layer is added again with 64 neurons. The dimension of the filter is set to 3x3. ReLU is used again as the activation function.

In the third layer, MaxPooling2D is used to reduce the dimensions of the feature map (down sampling), thereby speeding up computation because fewer parameters need to be updated and overfitting. Here is the formula for MaxPooling:

$$h^l_{xy} = max_{i=0,..,s,j=0,..,s} h^{l-1}_{(x+i)(y+j)}$$

The model object is added to the MaxPooling2D layer with the form of a 2×2 matrix as the minimum pixel loss and the precise region where the feature is allocated.

In the fourth layer, Dropout is used to prevent overfitting and speed up the learning process. Dropout works by randomly selecting neurons that will be ignored during training. Dropout itself is only used during model training and is not used when evaluating the model. In the proposed model, the probability value is set into 25% for each weight update cycle.

In the fifth layer, there is a flattening process with Flatten which functions to convert the pooling data in the form of a 2-dimensional array into one-dimensional single vector data.

Next, in the sixth layer, there is a Dense process that functions to add a fully connected layer. Dense itself has an operation that is:

$$\text{output} = \text{activation}(\text{dot}(\text{input}, \text{kernel}) + \text{bias})$$

where activation is the element-wise activation function passed as the activation argument, a kernel is a weights matrix created by the layer, and bias is a bias vector created by the layer (only applicable if use_bias is True).

In the proposed architecture, the dense model is set with total 128 nodes that must be in the hidden layer, in which the value is between the number of input nodes and output nodes, and we use the ReLU activation function.

On the seventh layer, again there is a Dropout process. This second Dropout uses a 50% chance in each weight update cycle.

Finally, in the eighth layer, Dense is used again to initialize the output layer, which is only two nodes because the dataset have 2 results i.e. IDC(+) and IDC(-), and we use the softmax activation function.

After the CNN model is complete, the next step is to compile the model with 3 parameters:

1. Loss parameters to determine the loss function, using keras.losses.categorical_crossentropy.
2. Optimizer parameters to define the stochastic gradient descent algorithm.
3. Metrics parameters to determine performance metrics, using 'accuracy'.

There are two optimizers that were considered to be tested in the model, which are Adadelta Optimizer and Adam Optimizer.

The first optimizer is Adadelta Optimizer, where Adadelta itself has the formula:

$$\theta_{t+1} = \theta_t + \Delta\theta_t$$

The second optimizer is Adam Optimizer, where Adam itself has the formula:

$$\Delta\theta_t = \frac{-\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t$$

Where:

- $\Delta\theta_t = -\frac{RMS[\Delta\theta]_{t-1}}{RMS[g]_t} g_t$
- $RMS[\Delta\theta]_t = \sqrt{E[\Delta\theta^2]_t + \epsilon}$
- $RMS[g]_t = \sqrt{E[g^2]_t + \epsilon}$
- $E[\Delta\theta^2]_t = \gamma E[\Delta\theta^2]_{t-1} + (1-\gamma)\Delta\theta_t^2$
- $E[g^2]_t = \gamma E[g^2]_{t-1} + (1-\gamma) g_t^2$
- $\hat{m}_t = \frac{m_t}{1-\beta_2^t}$
- $\hat{v}_t = \frac{v_t}{1-\beta_1^t}$

In CNN, the input vector is converted by a set of weights like a linear function, as shown in the following equation:

$$y = w.x + b$$

In this equation, y, x, and w refer to output, input, and weight, respectively. Meanwhile, the bias term, b, is a constant added to the linear equation to increase the flexibility of the model.

The activation function calculates the network output based on the input. This function will be applied to a linear equation. The activation function will decide whether neurons should be activated or not activated based on the number of neuron's weights and bias. There are several types of activation functions, such as sigmoid function, hyperbolic tangent function (Tanh), softmax function, softsign function, rectified linear unit (ReLU) function, and exponential linear units (ELUs) function. In this study, we used two types of activation function, ReLU function, and softmax function.

The ReLU function can be calculated with the following formula:

$$y = \max(0, x)$$

The softmax function can be calculated with the following formula:

$$y = \text{s}(xi) = \frac{e^{xi}}{\sum_{j=1}^{n} e^{xj}}$$

IV. Experimental Design

A. Dataset

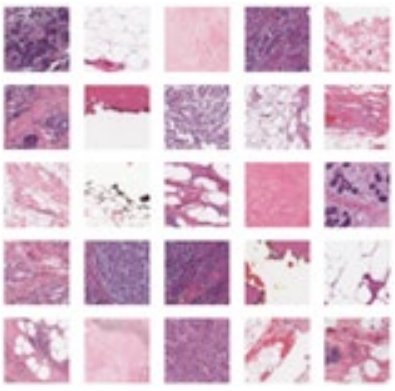

Figure (3). Histopathology images dataset of random patients' breasts

The histopathology images dataset used for this breast cancer detection is a public dataset which is belonging to Paul Mooney. We retrieve this dataset from Kaggle with

link https://www.kaggle.com/paultimothymooney/breast-histopathology-images/.

The original dataset consisted of 162 whole mount slide images of Breast Cancer (BCa) specimens scanned at 40x. From that, 277,524 patches of size 50 x 50 were extracted (198,738 IDC negative and 78,786 IDC positive). Each patch's file name is of the format: uxXyYclassC.png, where u is the patient ID (10253idx5), X is the x-coordinate of where this patch was cropped from, Y is the y-coordinate of where this patch was cropped from, and C indicates the class where 0 is non-IDC labeled as IDC(-) and 1 is IDC labeled as. The dataset was identified and categorized by breast cancer subtypes.

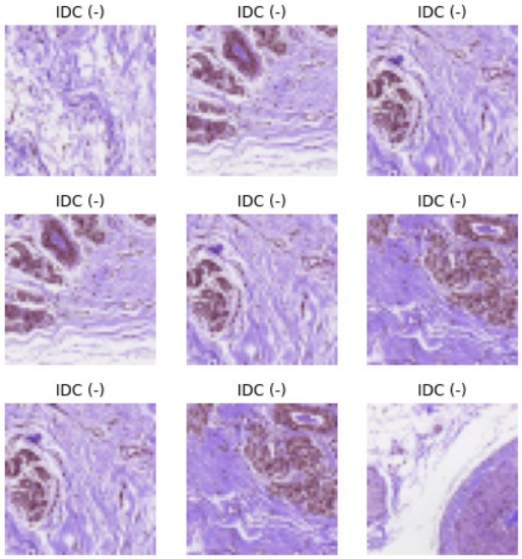

Figure (4). Histopathology images of random patients' breasts with non-IDC

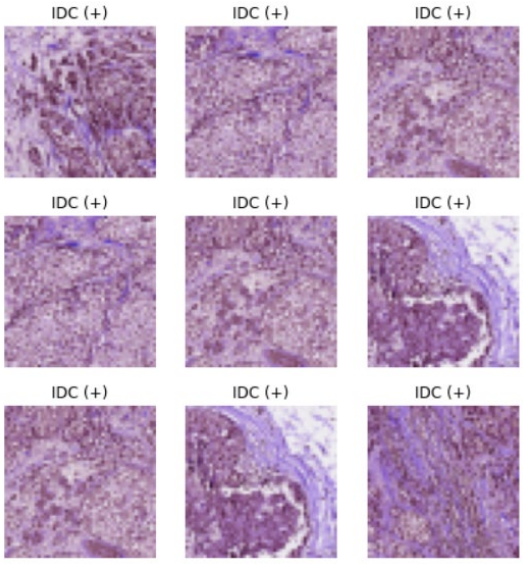

Figure (5). Histopathology images of random patients' breasts with IDC

Figure 4 shows the collection of photos of patients labeled as not suffering from Invasive Ductal Carcinoma, while Figure 5 shows random histopathology images of patient's breasts with Invasive Ductal Carcinoma breast cancer.

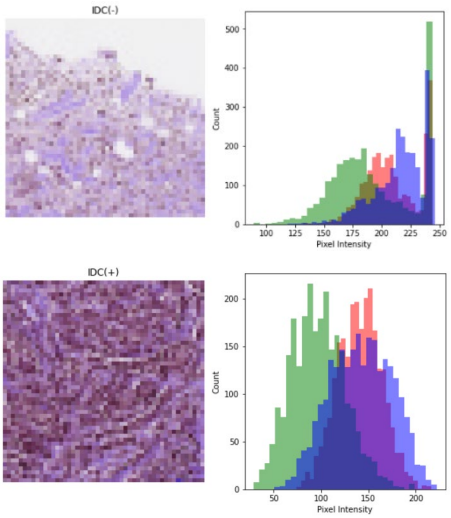

Figure (6). Histopathology image and its pixel intensity of random patient's breast

From the histopathology image of the patient's breast, it also can be plotted to see the pixel intensity of the RGB color, which will take 3 dimensions to establish the color of the image. From the comparison of both images, the image labeled with IDC(+) has a higher pixel intensity than IDC(-), where it can be seen that the dominance of purple is higher and darker in the IDC(+) histogram rather than in the IDC(-) histogram.

B. *Processed Dataset*

In this study, the used image dataset size is in 50x50 pixels. The process of labeling image data based on the file name is also carried out, if there is a class 1 name then it is labeled IDC(+) and if the name is class 0 then it is labeled IDC(-).

The total image dataset used is 40,000 images. The number of datasets is limited to avoid crashes because the training process takes up quite a lot of memory and Google Colab only provides a maximum limit of 12 GB of RAM. The following are specifications regarding the datasets that will be processed for the train and test the CNN model:

Total number of images: 40000
Number of IDC(-) Images: 29261
Number of IDC(+) Images: 10739
Percentage of positive images: 26.85%
Image shape (Width, Height, Channels): (50, 50, 3)

Furthermore, the dataset is divided into 2 types of datasets with a proportion of 80% for training datasets and 20% for testing datasets.

Because the amount of data based on the label is not balanced, where the number of images for the IDC(-) label is 29261 images and the number of images for the IDC(+) label is 10739 images, the Under-sampling method is carried out to balance the size of the number of images based on the label so that the final number of

images with labels is IDC(-) and IDC(+) each have 8641 images.

C. *Hyperparameter Exploration*

The CNN model architecture used is as defined in the proposed algorithm section. For hyperparameter exploration, experiments were carried out to change the optimizer, learning rate, batch size, and epoch with various variations.

For the optimizer, experiments were carried out on 2 types of optimizers, namely the Adadelta Optimizer and the Adam Optimizer. For the learning rate, we set values that vary from the default value of the optimizer, which is 1e-3, then we also try to set the values of 3e-3, 5e-3, 1e-4, 3e-4, and 7e-4. For the batch size, we tried to set various values, namely 32, 16, and 10. Finally, for the epoch value, in the initial model type training, we have not provided early stopping for the training model, but in the 3rd model up to the last model, We have implemented early stopping for the epochs that were carried out during training, where we monitored the loss value and determined patience from the number of epochs that did not improve to be stopped from training process with a value of 3.

D. *Performance Evaluation*

Because in this study a classification process was carried out to categorize whether the photo data stated that the patient had cancer or not, the evaluation of the model's performance will be carried out by looking at the following indicators:

A. Accuracy

Accuracy is expressed as the percentage of correct predictions for a model. The best accuracy is equal to 100%, while the worst accuracy is equal to 0%. Typically, accuracy is calculated by the following formula:

$$accuracy = \frac{\text{number of correct predictions}}{\text{total number of predictions}} x\ 100\%$$

Apart from that, accuracy can also be evaluated as positive and negative for binary classification with the following formula:

$$accuracy = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}\ x\ 100\%$$

Where:

1. TP is True Positives or the number of predictions of true in which a positive instance.
2. TN is True Negative or the number of the prediction of true that in the case is negative.
3. FP is False Positive or the number of predictions of false of negative instance.
4. FN is False Negative or the number of predictions of false that a case positive.

B. Precision

It measures the accuracy of the model and represents the ratio of carcinoma images classified accurately from a composite of class images that predict the same. And it can be calculated with the following formula:

$$precision = \frac{\text{TP}}{\text{TP} + \text{FP}}\ x\ 100\%$$

C. Recall

Recall also called "sensitivity" computes the completeness of a model. It represents the ratio of images accurately classified as carcinoma out of the total number of carcinoma images. And it can be calculated with the following formula:

$$recall = \frac{\text{TP}}{\text{TP} + \text{FN}}\ x\ 100\%$$

D. F1 Score

It shows the harmonic averages of precision and gain and is typically used to optimize a model toward precision or gain. And it can be calculated with the following formula:

$$F1\ score = \frac{2 * (\text{Recall} * \text{Precision})}{\text{Recall} + \text{Precision}}$$

E. EXPERIMENTAL RESULT

This study experiment was carried out using the help of Google Collaboratory to execute the code and using libraries from Python. The CNN method used produces quite good results in this experiment.

No	Optimizer	Learning Rate	Batch Size	Epoch	Accuracy	Precision	Recall	F1 score
1	Adadelta	0.001	32	50/50	78,01%	0.78	0.78	0.78
2	Adadelta	0.001	16	50/50	78,86%	0.79	0.79	0.79
3	Adadelta	0.001	10	54/100	79,54%	0.80	0.80	0.80
4	Adam	0.001	10	18/100	81,31%	0.83	0.81	0.81
5	Adam	0.005	10	6/100	50%	0.25	0.50	0.33
6	Adadelta	0.005	10	55/100	82,59%	0.83	0.83	0.83
7	Adadelta	0.003	10	33/500	80,63%	0.81	0.81	0.81
8	Adam	0.0007	10	19/500	85,11%	0.85	0.85	0.85
9	Adam	0.0005	10	30/500	86,70%	0.87	0.87	0.87
10	Adam	0.0003	10	29/500	86,03%	0.86	0.86	0.86

Table (1). Hyperparameter Exploration and the Experimental Result

MODEL 1

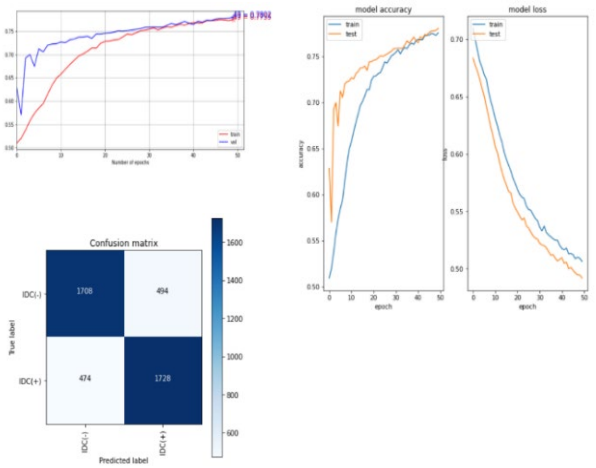

Figure (7). Learning curve, model accuracy, model loss, and confusion matrix plot of model type 1

MODEL 2

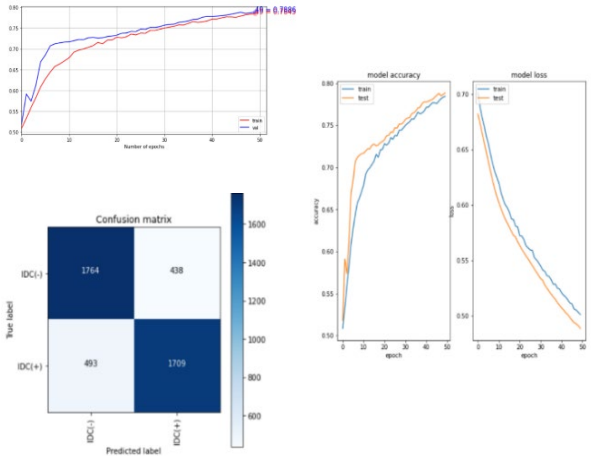

Figure (8). Learning curve, model accuracy, model loss, and confusion matrix plot of model type 2

MODEL 3

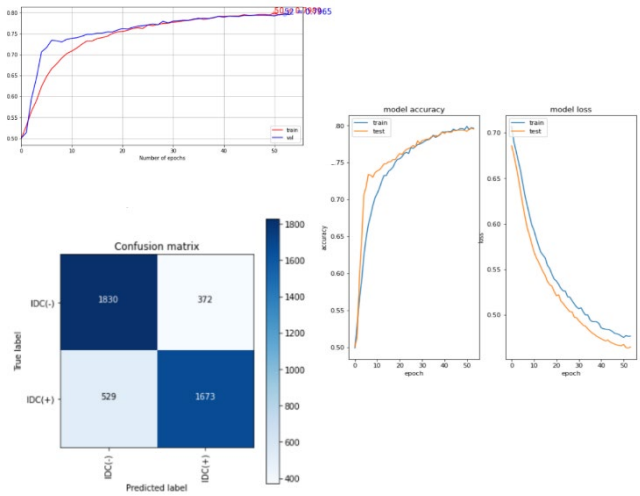

Figure (9). Learning curve, model accuracy, model loss, and confusion matrix plot of model type 3

MODEL 4

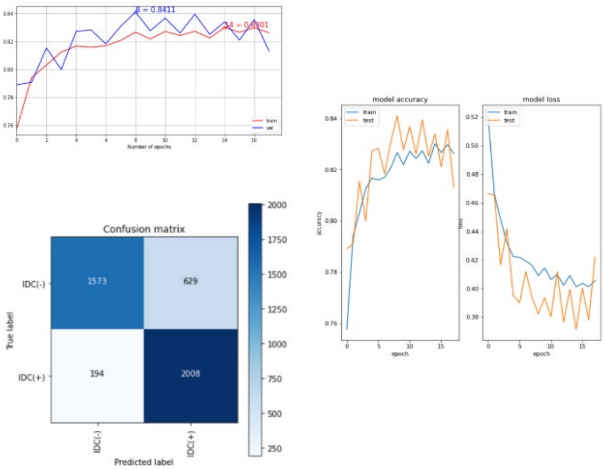

Figure (10). Learning curve, model accuracy, model loss, and confusion matrix plot of model type

MODEL 5

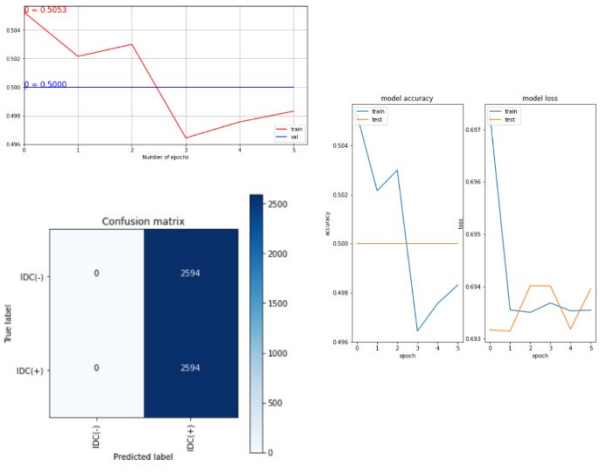

Figure (11). Learning curve, model accuracy, model loss, and confusion matrix plot of model type 5

MODEL 6

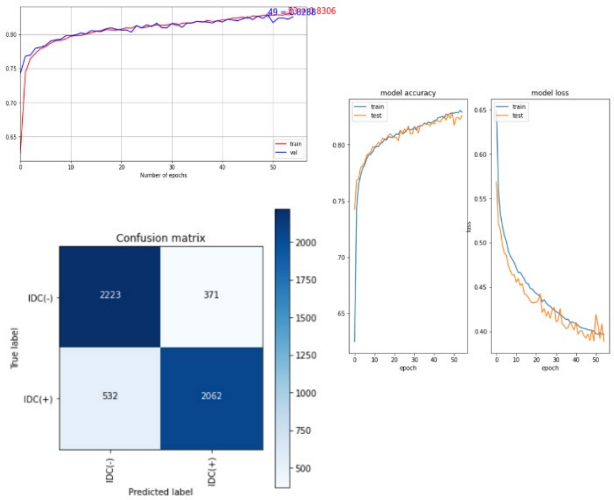

Figure (12). Learning curve, model accuracy, model loss, and confusion matrix plot of model type 6

MODEL 7

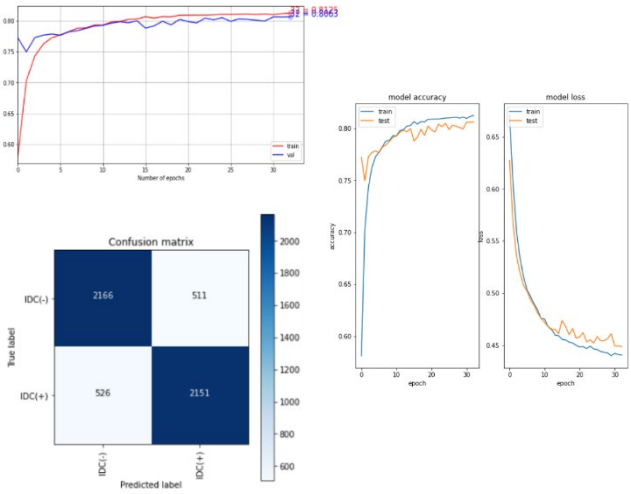

Figure (13). Learning curve, model accuracy, model loss, and confusion matrix plot of model type 7

MODEL 8

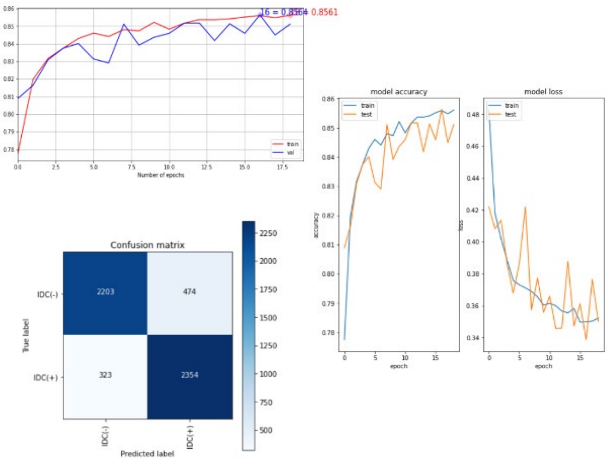

Figure (14). Learning curve, model accuracy, model loss, and confusion matrix plot of model type 8

MODEL 9

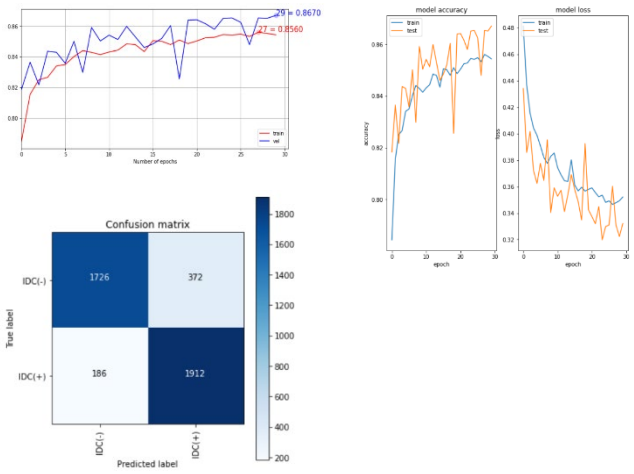

Figure (15). Learning curve, model accuracy, model loss, and confusion matrix plot of model type 9

MODEL 10

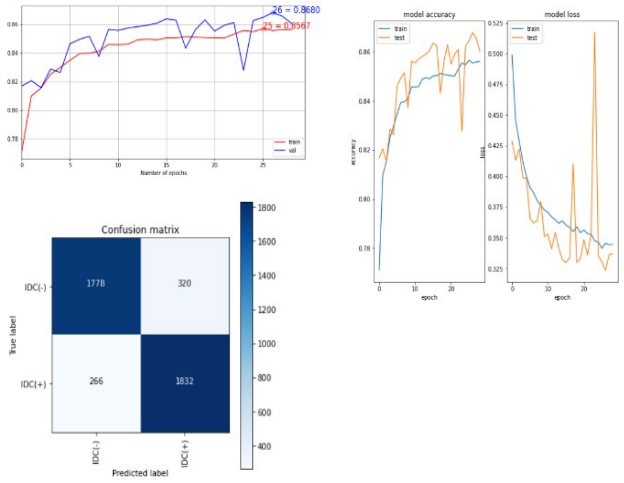

Figure (16). Learning curve, model accuracy, model loss, and confusion matrix plot of model type 10

Based on this experiment, we can see that batch size can affect the accuracy of a model. Batch size is one of the most important hyperparameters to suit a deep learning system. It should be noted that the larger the batch size will produce bad results. For example, models 1, 2, and 3, use the same optimizer, namely Adadelta, and the same learning rate of 0.001. What distinguishes the three of them is the batch size. Model 1 uses a batch size of 32, model 2 uses a batch size of 16, and model 3 uses a batch size of 10. The resulting accuracy increases as the batch size decreases.

Next, the proposed architecture tries two types of optimizers. The first optimizer is Adadelta and the second is the Adam optimizer. For example, in model 3 that use the Adadelta optimizer, and in model 4 that use the Adam optimizer with the same learning rate, which is 0.001, and the same batch size which is 10. The experimental result shows that the model with the Adam optimizer has better accuracy than a model with Adadelta optimizer. However, for the model 3 with Adadelta optimizer, the plots of learning curve, model accuracy, and model loss are showing more stable plots rather than in model 4 that uses Adam optimizer. The stable accuracy plot indicates that the training is quite good, not underfitting or overfitting. The stable model loss plot represents that the used learning rate is good. The confusion matrix of model 3 also shows more balanced value in the False Positive and True Negative cells, while in model 4, the value of False Positive is much bigger than the value of True Negative, which implies that this model will do more misdiagnose people that actually IDC(-) but are mislabeled as IDC(+).

The exploration has also done by varying the learning rate values. There are 6 different learning rates that were tested, which are 0.0003, 0.0005, 0.0007, 0.001, 0.003, and 0.005. In models 4, 5, 8, 9, and 10, all three use the Adam optimizer and batch size 10, the difference between them is the learning rate. In model 4, using a medium learning rate of 0.001. In model 5, using a too large learning rate of 0.005. In model 8 using a learning rate of 0.0007. In model 9 using a learning rate of 0.0005. While in model 10 using the smallest learning rate of 0.0003. From the comparison of the five models, model 9 gets the best accuracy results. We can see that for Adam optimizer, the smaller the learning rate, the greater the accuracy, and vice versa. You should not use a learning rate that is too small as seen in model 10, because it will reduce the accuracy value. However, when using the Adadelta optimizer on models 3, 6, and 7 with the same batch size of 10 and varying learning rates, there is a significant difference between the Adadelta and Adam optimizers. It can be seen in model 3, using a learning rate of 0.001. In model 6, using a learning rate of 0.005. And model 7 uses a learning rate of 0.003. From the comparison of the three models, model 6 with a learning rate of 0.005 gets the best accuracy. For the use of the Adadelta optimizer, the greater the learning rate used, the better the accuracy obtained. This is of course inversely proportional to the Adam optimizer, where the model is getting better when the learning rate value of model 6 is increased and model 9 is decreased.

From the explorations and explanations above, also considering the most acceptable model, then the model 9 is chosen. The model 9 with Adam Optimizer, learning rate 0.0005, batch size 10, and the epoch that stops at the value of 30, has resulted an accuracy of 86.70%, with a balanced precision, recall, and f1 score value of 0.87. Although the learning curve of model 9 is more unstable than model 6 that using Adadelta optimizer, but ratio of the True Negative value in model 9 is much smaller than model 6, so that diagnostic errors where patients who should be positive cancer but are declared negative cancer can be minimized.

In this study, we realize that there are still some shortcomings, such as the type of dataset that is generated randomly so that every time all code is run from scratch, the dataset that will be processed also changes. Then the number of datasets that we had to limit was only up to 40000 histopathological images due to the limited memory

specification of 12 GB which was used when this research was conducted.

F. Conclusion

Breast cancer detection is a challenging problem as it is the most popular and known to be dangerous. Breast cancer develops every year and is less likely to recover from this disease. We try to detect the presence of breast cancer on histopathological images so that breast cancer patients can be treated immediately when detected. For this reason, we created a breast cancer detection model using the Convolutional Neural Network (CNN) method. The use of this CNN method in the detection of breast cancer produces quite good results. We also did some experiments to find out which one works better by varying the existing hyperparameters. The highest accuracy result that we achieved with this method was 86.70% using the Adam optimizer along with a learning rate of 0.0005, batch size 10, and also epoch 500 which was affected by early stopping at epoch 30. However, there are still some shortcomings in this study, which are the inconsistent dataset when all code is rerun and the limited number of datasets processed due to memory limitation.

References

[1] Shiva Shankar, R., Murthy, K. V. S., & Someswara Rao, C. (2020). Breast Cancer Disease Prediction With Recurrent Neural Networks (RNN). International Journal of Industrial Engineering & Production Research, 31(3), 379-386. http://ijiepr.iust.ac.ir/files/site1/user_files_2hops2/appaji-A-10-1605-1-2d84cb2.pdf

[2] Collignon, J., Lousberg, L., Schroeder, H., & Jerusalem, G. (2016). Triple-negative breast cancer: treatment challenges and solutions. Breast Cancer: Targets and Therapy, 8, 93. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4881925/

[3] Ali, E. E. E., & Feng, W. Z. (2016). Breast cancer classification using support vector machine and neural network. International Journal of Science and Research, 5(3), 1-6. https://pdfs.semanticscholar.org/1b3a/5a698ec834043eb850edf1daa0ebb47cc14a.pdf

[4] DeSantis, C. E., Ma, J., Gaudet, M. M., Newman, L. A., Miller, K. D., Goding Sauer, A., ... & Siegel, R. L. (2019). Breast cancer statistics, 2019. CA: a cancer journal for clinicians, 69(6), 438-451. https://acsjournals.onlinelibrary.wiley.com/doi/full/10.3322/caac.21583?fbclid=IwAR3lclBZrF3EvvOgE-v9JIPdSyLzijq5XvbNZGDkvmrCDF7iqnZG2SfSQJ

[5] Mardela, A. P., Maneewat, K., & Sangchan, H. (2017). Breast cancer awareness among Indonesian women at moderate-to-high risk. Nursing & health sciences, 19(3), 301-306. https://onlinelibrary.wiley.com/doi/pdfdirect/10.1111/nhs.12345

[6] Stathonikos, N., Nguyen, T. Q., Spoto, C. P., Verdaasdonk, M. A., & van Diest, P. J. (2019). Being fully digital: perspective of a Dutch academic pathology laboratory. Histopathology, 75(5), 621-635. https://onlinelibrary.wiley.com/doi/pdfdirect/10.1111/his.13953

[7] Ha, R., Mutasa, S., Karcich, J., Gupta, N., Van Sant, E. P., Nemer, J., ... & Jambawalikar, S. (2019). Predicting breast cancer molecular subtype with MRI dataset utilizing convolutional neural network algorithm. Journal of digital imaging, 32(2), 276-282. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6456631/

[8] Chen, M., Zhang, B., Topatana, W., Cao, J., Zhu, H., Juengpanich, S., ... & Cai, X. (2020). Classification and mutation prediction based on histopathology H&E images in liver cancer using deep learning. NPJ precision oncology, 4(1), 1-7. https://www.nature.com/articles/s41698-020-0120-3

[9] Ruder, S. (2016, November 26). *Optimization for Deep Learning*. Optimization for Deep Learning. https://www.slideshare.net/SebastianRuder/optimization-for-deep-learning

[10] Karpathy, A. (). CS231n Convolutional Neural Network for Visual Recognition. https://cs231n.github.io/neural-networks-3/