Predicting Medical Images Using Convolutional Neural Network

Thesis for the Degree of Doctor of Philosophy (PhD)

by Oktavian Abraham Lantang

Supervisor: Dr. György Terdik

UNIVERSITY OF DEBRECEN

Doctoral Council of Natural Sciences and Information Technology

Doctoral School of Informatics

Debrecen, 2022

Hereby I declare that I prepared this thesis within the Doctoral Council of Natural Sciences and Information Technology, Doctoral School of Informatics, University of Debrecen in order to obtain a PhD Degree in Informatics at Debrecen University.

The results published in the thesis are not reported in any other PhD theses.

Debrecen, 2022

.....
signature of the candidate

Hereby I confirm that Oktavian Abraham Lantang candidate conducted his studies with my supervision within the Applied Information Technology and its Theoretical Background Doctoral Program of the Doctoral School of Informatics between 2016 - 2022. The independent studies and research work of the candidate significantly contributed to the results published in the thesis.
I also declare that the results published in the thesis are not reported in any other theses.
I support the acceptance of the thesis.

Debrecen, 2022

.....
signature of the supervisor

Predicting Medical Images Using Convolutional Neural Network

Dissertation submitted in partial fulfilment of the requirement for the doctoral (PhD) degree in Informatics

Written by Oktavian Abraham Lantang
certified computer science lecturer

Prepared in the framework of the doctoral school of informatics of the University of Debrecen
(Applied information technology and its theoretical background programme)

Dissertation advisor: Dr. György Terdik

The official opponents of the dissertation:

Dr. ....
Dr. ....
Dr. ....

The evaluation committee:

chairperson: Dr. ....
members: Dr. ....
Dr. ....
Dr. ....
Dr. ....

The date of the dissertation defence: 2022

Abstract

Cancer is one of the non-contagious diseases that causes the most significant number of human deaths. The time it takes for cancer cells to be identified in a patient's body significantly impacts how easily they can be treated. An automatic screening method using computer-aided diagnostic (CAD) is one way of early cancer detection. This dissertation proposes several models that can be adopted as automatic screening methods. The first model is a convolutional-based single neural network built by adopting the Visual Geometry Group (VGG) module. The second model is an ensemble model based on a voting system built by combining three single networks from scratch by adopting three well-known modules: VGG, Inception, and Residual Network modules. The last model is an ensemble model based on interconnected networks. Unlike the previous ensemble model, this model does not use a voting method in decision making but trains all three networks in an extensive linked network to make a single final decision. Furthermore, the success of each model, also the benefits and drawbacks of it are presented.

Contents

List of Figures

List of Tables

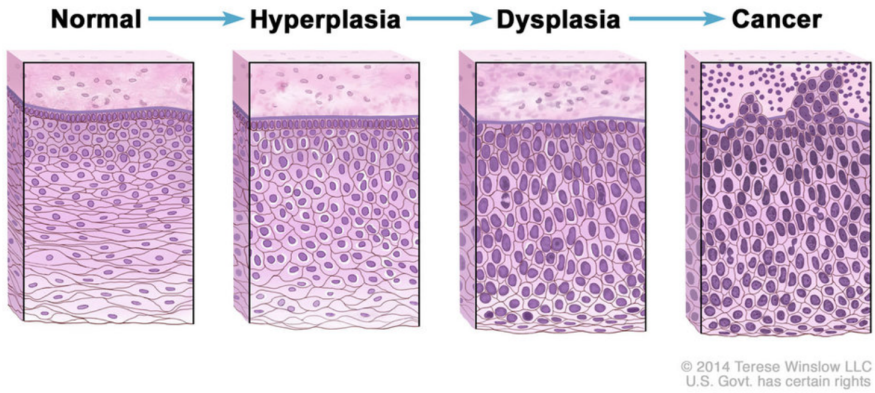
1. Introduction

1.1. Motivation

Cancer is one the most lethal diseases in the world at the present time. Cancer itself is categorized as a non-communicable disease. Through a study, the World Health Organization (WHO) stated that cancer kills humans during productive age or before 70 years [5]. Cancer can be defined as hundreds of types of diseases with different characteristics from one another. Human cells, which are abundant, can reproduce and are dependent on one another. The human body can control these cells' proliferation in normal conditions to maintain their shape and size. However, cancer cells interfere with this growth by moving from one place to another and change shape and size beyond the control of the human metabolism [56].

Clinical diagnosis is necessary to be done by doctors predominantly before confirming whether a patient has cancer or not. Furthermore, the patient is advised to do a pathological examination by an expert using a microscope. Presently, with the availability of Whole Slide Image Scanner, there has been a significant improvement in the quality and quantity of the cancer diagnosis process due to its ability to detect variables in a more comprehensive histopathology image [35]. Another advantage of this tool

is its ability to record appropriately all biological processes in the human body such as apoptosis, angiogenesis, and metastasis [57]. The process of digitizing by the Whole Slide Image Scanner data resulted in an explosion of large amounts of data. Thus, the availability of sufficient data can be utilized to implement deep learning by developing a Convolutional Neural Network (CNN) model [48]. Figure 1.1 shows how cancer arises.



Source: https://www.teresewinslow.com/cellular-scientific/ihj90o9dtfaen5y6klyz2c97o92vr7

Figure 1.1: How cancer arises

The CNN model is reliable for predicting image data. In this previous study [25], They developed a CNN model called AlexNet that consists of two-layer blocks. The first block contains five stack convolution layers, while the second block consists of three fully connected layers. This model got satisfactory results in the competition held by ImageNet, by analyzing 1.2 million images, this

model got an error rate of 16.4%. Meanwhile, the lower error rate of 15.3% was achieved when they modified the architecture to 7 CNN. The two error rates represent the top five predicted labels.

Visual Geometry Group (VGG) is one of the successful CNN architectures in the 2014 ILSVRC ImageNet competition. By combining the idea of definition, convolution block repetition, polling layers, and small filters, VGG is developed into a model with efficient and accurate performance. VGG was produced in two models, namely 16 and 19 layers. In particular, the 19 layers VGG was reported able to reach an error rate of 6.8% for the top five predicted labels and 23.7% for the top first predicted labels [49].

The main purpose of using CNN is to extract features from an image. The problem then arises due to the presence of the various positions of the information in the image. In other words, the information from the image does not always exist at the same point, sometimes we may find the desired information in the edge of an image and may only occupy a small part of the image area. These conditions require extra work in determining a suitable filter size for CNN. If we use a too large filter, the disbursement of information will be more global and lead to the high computation costs. Meanwhile, if we use a too-small filter, the process of searching for information is more local, which may lead to the possibility of losing the desired information. To solve this problem, the Inception

architecture came up with the idea of installing filters of different sizes at one convolution level and then combining the outputs. The purpose of this integration is to reduce the computation costs without compromising model performance. By installing some of these filters, this architecture will look wider than other architectures [52].

Another idea starts when there is the fact that too deep and complex architecture ruins the model's performance. Thus a concept was developed to divide the convolution into several blocks, and each block has a shortcut for bypassing. This architecture produces a model known as the Residual Network (ResNet). ResNet has several types, varied between the smallest architecture with 34 layers to the most complex with 152 layers. From the training and validation process using the ImageNet dataset, ResNet achieved a 19.38% error rate for the top first predicted labels and a 4.49% error rate for the top five predicted labels [17].

Understanding the fact that to distinguish histopathology images containing cancer cells and not containing cancer cells requires an expert's ability and supported by the fact that the development of machine learning methods has reported satisfactory achievements in classifying image data, this dissertation focuses on implementing artificial intelligence such as a pathologist in classifying human histopathology images by utilizing machine learning

methods, especially convolutional neural networks.

1.2. Contribution and main structure of the dissertation

This dissertation's main idea has been published in both proceedings and journals in the form of articles that will be discussed later. This work aims to optimize the CNN model for predicting cancer cells in human histopathology images. The first experiment was run by developing a VGG module. The convolution layers in the VGG model were re-implemented with several adjustments, including the number of layers, convolution channels, filters, multilayer perceptron (MLP), and convolution blocks. The histopathology images in the dataset were trained using a model developed to detect cancer cells.

The second experiment focused on ensemble-based CNN development. Ensemble-based CNN is a combination of two or more sub-models in order to get better performance. The proposed ensemble model was a combination of three sub-models built from scratch by implementing three well-known CNN modules, namely the VGG module, the Inception module, and the ResNet module.

The last experiment focused on the efficiency of the ensemble model. If the previous work ensemble members were trained

separately, and the final result was determined by voting, the ensemble members were trained simultaneously in an interconnected model in this work. Furthermore, in the first ensemble model, each sub-model weight were determined by the user, but in the second ensemble model, the weighting was part of the training process. The interconnected model determined the appropriate weight for each sub-model. Thus, the sub-model with better performance will possess a bigger weight than the other model.

Furthermore, the structure of this dissertation is arranged as follows:

1. Introduction

   This chapter explains the motivation, background, issues discussed, related publications, and the dissertation's writing structure.

2. Literature Review

   This chapter discusses the methods used and the related work of this dissertation.

3. Results and Discussion

   This part contains three chapters that explained the results obtained from the experiment and its comprehensive discussion.

4. Conclusion

   This chapter presents the conclusions of all the work achieved in this dissertation.

2. Literature Review

2.1. Dataset

The data used in this dissertation was the PatchCam dataset. PatchCam is a human histopathology image produced by the Whole Slide Digital Scanner. This dataset was derived from the Camelyon16 dataset [2, 55]. Camelyon16 itself was a whole-slide image of the lymph node section, which was then broken down into 220,025 pathology images with a size of 92x92 pixels. PatchCam was used in the Kaggle Histopathology Cancer Detection competition. Figure 2.1 is an example of a cancer image, and Figure 2.2 is an example of a non-cancer image from the PatchCam dataset.

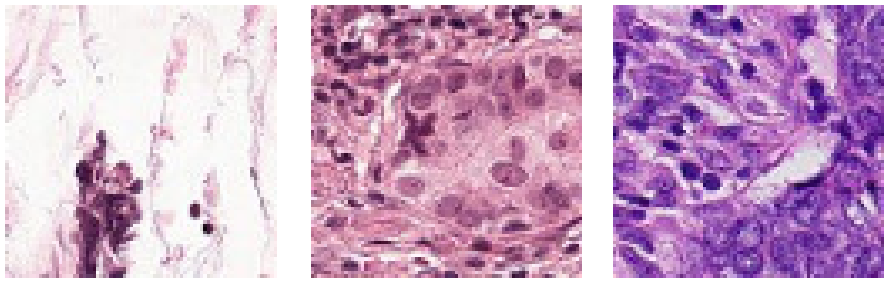


Figure 2.1: Cancerous images.

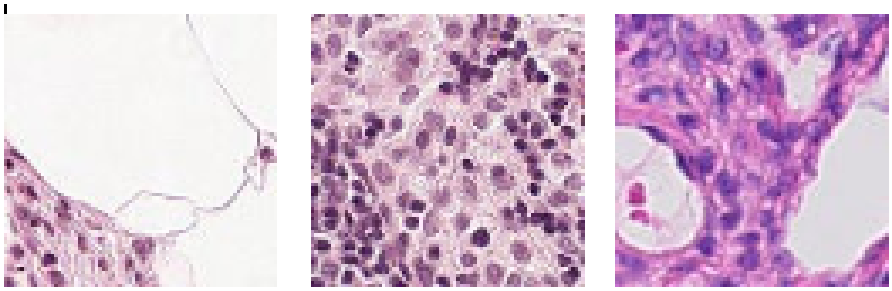


Figure 2.2: Non-cancerous images.

2.2. Convolutional Neural Network

Before we discuss the suggested architecture, it is advised to understand the CNN architecture components beforehand. In general, the CNN architecture is divided into two large blocks. The first block is the feature extraction layer, in which the encoding process occurs from the image into features. Specifically, these features are the numbers that represent the image. If we break this block up again, two layers work in it: the Convolution layers and the Pooling layers.

The convolutional network, known as Convolutional Neural Network (CNN) [30], is a type of neural network that can process image data [8, 47, 19, 44, 36, 21, 51, 41, 50] and analyze medical images [53, 1, 6, 16, 15, 32, 26, 59, 39, 23, 10, 34]. The terminology of convolutional neural networks refers to the use of a mathematical operation called convolution. Meanwhile, the mathematical process

of convolution is a type of matrix multiplication tailored to the needs of the neural network [12]. Furthermore, it is explained that one of the implementations of the convolution operation is for two-dimensional image processing. This process can be seen in [12] as equation (2.1).

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n), \quad (2.1)$$

It is explained that when we have a two-dimensional, I input image and a two-dimensional K kernel, the output value S is calculated by adding up each product of the multiplication of the pixel value in I with K. Since the convolution is commutative, we can also write the process as equation (2.2):

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n), \quad (2.2)$$

For example, Figure 2.3 represents the input image a with kernel b, then one of the convolution processes of a and b will produce an output $S(1,1) = (U1,1 * K1,1) + (U1,2 * K1,2) + (U1,3 * K1,3) + (U2,1 * K2,1) + (U2,2 * K2,2) + (U2,3 * K2,3)$. From Figure 2.3, it can be understood that to complete the full convolution process, there will be pixel shifting in the input image a to be multiplied by the kernel b. This shifting is called Stride [7]. If

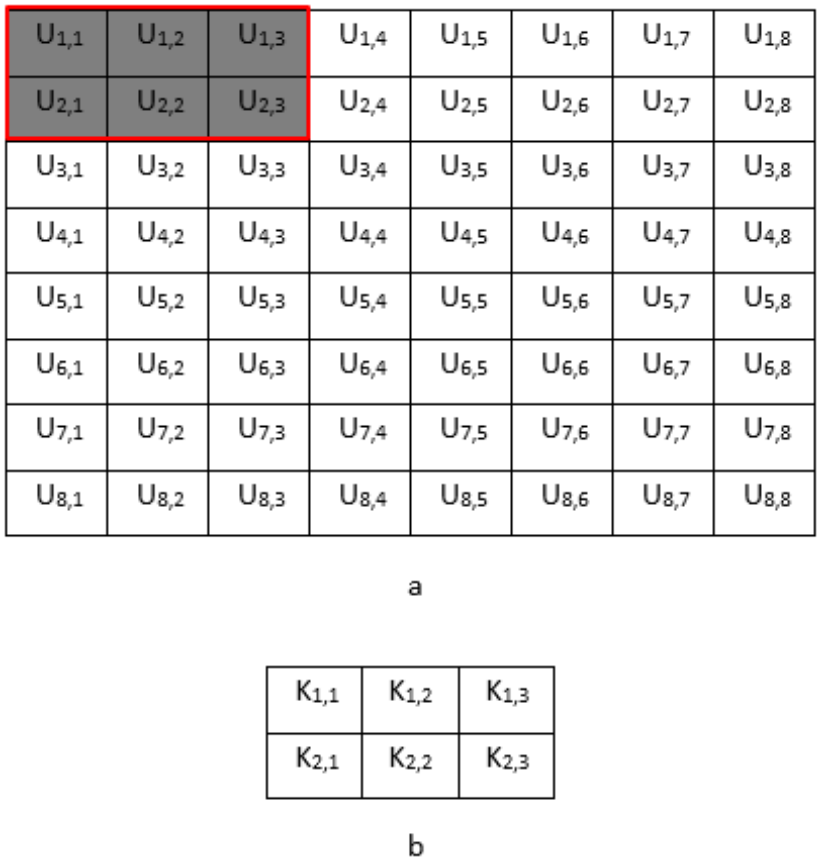


Figure 2.3: Convolution process.

the Stride value is one, then the shifting of the red box is one pixel horizontally and vertically. The smaller the Stride value, the more detailed information obtained. However, it should be noted that the smaller the pixel shifting will also impact the computation cost significantly. Apart from that, in fact, the output size will always be smaller than the input size. As the output will be used as in-

put in the following convolution process, the output's overall size will continuously decrease. This might have an impact of missing important information from the input. In order to keep the output size from decreasing drastically, a method called Padding is used [7]. Padding manipulates the input size by adding the 0 pixel value vertically and horizontally. The padding value represents the number of pixels to be added. Thus, to calculate the dimensions of the output, the following formula can be used.

$$Output = \frac{W - N + 2P}{S} + 1, \tag{2.3}$$

Where;

W = Row / Column input size,

N = Row / Column input size,

P = Padding and,

S = Stride.

Pooling layers are another type of layer in a convolution block, which is basically a filter with a certain Stride value to poll the feature map. There are two types of polls often used: Max-Pooling and Average-Pooling. Max-Pooling will choose the maximum value on the feature map while the Average-Pooling layer will average the value from the feature map.

In the next block, the resulting feature map will be continued

on the fully-connected layer (FC), which is a multilayer-perceptron (MLP) in the form of a neural network in general with a Neuron and Activation function [12, 35, 40]. At this stage, there are dimensional differences between the feature map and the MLP. The feature map, which is the convolution process's output, is a multi-dimensional matrix, while MLP is a vector. To change the feature map dimensions to a vector, a Flatten layer between convolution blocks and MLP is needed. In this dissertation, two types of activation functions were used, namely Softmax and Rectified Linear Unit (ReLU). The Softmax function has been very often used in deep learning implementation [40]. By providing a probability value for each class, Softmax was not only used in multi-classification problems but also for binary classes. Thus, the softmax function can be written as follow:

$$f(x_i) = \frac{exp(x_i)}{\sum_j exp(x_j)}, \tag{2.4}$$

ReLU is an Activation function that provides a threshold for values less than zero. In other words, this function sets all values less than zero to zero. Thus the ReLU function can be written as:

$$f(x) = max(0, x) = \begin{cases} x_i, & \text{if } x_i \geq 0 \\ 0, & \text{if } x_i < 0 \end{cases} \tag{2.5}$$

Currently, ReLU is the fastest activation function in the deep learn-

ing process [31] and has proven to be the most widely used [43], because compare to the other activation functions, ReLU is proven to have better performance [60, 9]. ReLU can quickly perform the learning process because it does not calculate exponential and division processes like other activation functions [60].

2.3. Training parameters

In the model training process using a dataset, several parameters need to be set. Generally, it can be explained that an algorithm is required to train the CNN model. One of the common iterative learning algorithms that are used is Stochastic Gradient Descent (SGD) [3, 4]. There are at least two hyperparameters of SGD that need to be understood in order to be appropriately implemented. The two hyperparameters are Batches and Epochs.

As previously discussed, the deep learning process utilizes artificial neural networks to train the model. The SGD is an optimization algorithm for carrying out the learning process. The optimization process refers to a process of minimizing the loss function, ensuing to the closer resulting model to the desired model. Because the optimization process is carried out repeatedly until it gets the smallest loss value, the number of epochs needs to be defined to determine how many times the learning algorithm will work on the

entire dataset. Thus, epochs is the hyperparameter of SGD, which determines how many iterations the dataset will be trained on. The number of epochs varies, usually set to 10, 100 or even 1000. If the epochs is set to 1, each sample in the dataset has one chance to update model's parameter. Batches or batch size is a hyperparameter of SGD that determines how many samples will be worked when updating the model's parameters[1]

Several algorithm models with the gradient descent approach have been developed, one of them is Adam optimizers [24]. The Adam optimization algorithm is an extension of SGD and has been widely used in deep learning implementations. Adam adopted the advantages of the Adaptive Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp). Some of the advantages of using Adam optimizers, This method is easy to implement, it requires low computation costs, is requires little memory space, is invariant to scale the diagonals of the gradient, is suitable for solving big data problems or many parameters, ideal for non-stationary purposes and solving sparse gradient problems.

[1] https://keras.io/api/models/model_training_apis/.

2.4. Preprocessing and augmentation

In the deep learning process, a large amount of data is needed. More data available, the better the model that will be produced. The process of augmentation in this work referred to [38]. This process included several stages of manipulation, intending to duplicate the image. Although the duplication was made, the duplicated image was not identical to the original image. Before discussing augmentation any further, we need to see that the main reason for using data Augmentation, was because the model was overfitting, which refers to the low validation value compared to a model's accuracy value. A model can experience overfitting due to the lack of inside data learning process. Luckily in Keras[2], we can use a simple data augmentation method. In this work, we utilized several methods of data augmentation. These methods including; Rotation, Shearing, Shifting, Zoom, and Flipping.

2.5. Pre-trained model

The Pre-trained model is a model that has previously improved, and it is proven to have a good accuracy value in doing predictive work. One of the pre-trained models which have suc-

[2] https://keras.io/api/preprocessing/image/

cessfully performed the predictive work is VGG19 [49]. VGG19 is the CNN developed by Oxford University and entered the competition ImageNet Challenge ILSVRC in 2014. By relying on 19 layers, VGG19 achieved an error rate of 7.5% when validating data and a 7,3% error rate when testing data. The architecture of VGG19 included 16-layer convolution and 3 fully connected layers. Besides, 16 layers was arranged in such a way within 5 convolution blocks. Each block ends with Max-Pooling layer, which aimed to reduce the dimensions of the convolution. Before passing through fully connected layers, the dimensions of convolution must be changed first. For that, flatten layers were used in this architecture. The fully connected layers previously mentioned was a Multilayer Perceptron (MLP) consisting of three fully connected layers. The first two fully connected layers used the ReLu activation function, while the latter used softmax 1000 class according to the competition's problem classification.

The other trained model which successfully doing the classification and prediction of image data was the InceptionV3 [52]. This model is an extension of the successful GoogLeNet model at the ImageNet competition. Module inception concept developed based on the fact that image data have a unique characteristic. An image may contain multiple objects, and not all objects constitute an important part of the data set. Furthermore, the object's posi-

tion is not always in the center of the image, sometimes on the edge picture. This condition is a challenge in designing the model. The use of large filters in the model design will lead to higher computational costs. On the other hand, the utilization of small filters leads to missing important information from the picture. The Inception module provides a solution using different filter sizes utilization in one level terminated with the concatenation process. Using multiple filters with different sizes on one level, causing the architecture to look physically wider instead of deeper. The single InceptionV3 model reported an error rate of 18.77% for top-1 label prediction and a 4.2% error rate for the top-5 predicted label.

The pre-trained ResNet [17] model was an architecture that develops shortcut concepts for doing a bypass. The idea of this model was to group multiple convolution layers in a block. Therefore, this model entirely looks like several collections of convolution blocks. Further, each block will be experiencing a bypass with a shortcut of the same convolution layer on every block. In addition to the form of blocks, this architectural convolution also appears to have a repetitive bypass process. The results achieved can explain the weaknesses of a deep architecture that in fact can damage the accuracy of the model. ResNet has several layer variants ranging from 34 layers to 152. When tested with an ImageNet dataset, the best results shown by ResNet-152 with the result of 19.35% error

rate on the top-1 predicted label and 4.49% the error rate on the top 5 predicted labels.

3. Single CNN to Classify Medical Images

3.1. Network architecture

In this chapter, we report our work involved a single CNN architecture in predicting cancer on histopathological images [29]. This architecture was designed by adopting the VGG module. Referring to the architecture, VGG stacks convolution layers. Based on this idea, we stacked eight convolution layers, which were two 64 channel of convolution, two 128 channel of convolution, two 256 channel of convolution, and two 512 channel of convolution. Later, every two of these convolutions, a Max-pooling layer was inserted. Thus, four convolution blocks were formed, filled with two convolution layers and a max-pooling layer in each block. The next block was the Multilayer Perceptron (MLP), a one-dimension vector. This requires the information that took shape multidimensional during the convolution process to be converted before entering the MLP block. This process involved a flatten layer which converts the multidimensional matrix to be a vector. After that, we installed two of the Dense layers with 64 channels and ended with the Dense layer two channel according to the problem in our dataset, the binary classification. We utilized the ReLu activation

function on all layers except for the last layer that used the Sigmoid activation function. To avoid overfitting conditions, we also put two Dropout 0.5 after the 64 channel Dense layer. For more details, we show the architecture in Figure 3.1.

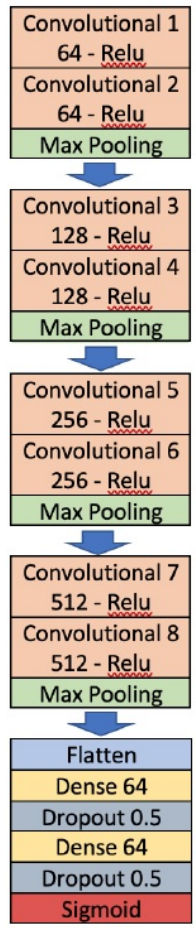


Figure 3.1: Architecture single CNN.

3.2. Training process

In general, two part of training process was included. The first part started from the input process and four convolution blocks ended with the Max pooling layer. This process was coding an image into a number which represents the image. The results of the coding can be seen in a Feature Map as shown in Figure 3.2. In this figure, part a was the input image given to the model while

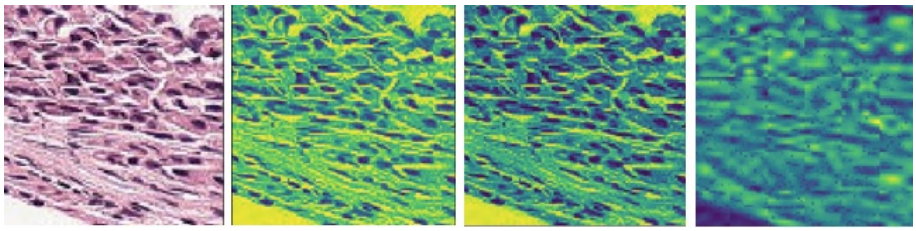


Figure 3.2: Feature extraction stage.

part b-d was the extraction feature of the input image generated by the model. Since each layer will generate a Feature Map, the first block of the architecture we designed will generate a Feature Map as shown in Figure 3.3. As we can see in Figure 3.3, it consisted of three layers representing the three layers in the first block in Figure 3.2. The first feature map represented the first 64 channel convolution layer, the second feature map represented the second 64 channel convolution layer, while the third feature map represented the Max pooling layer of the first convolution block.

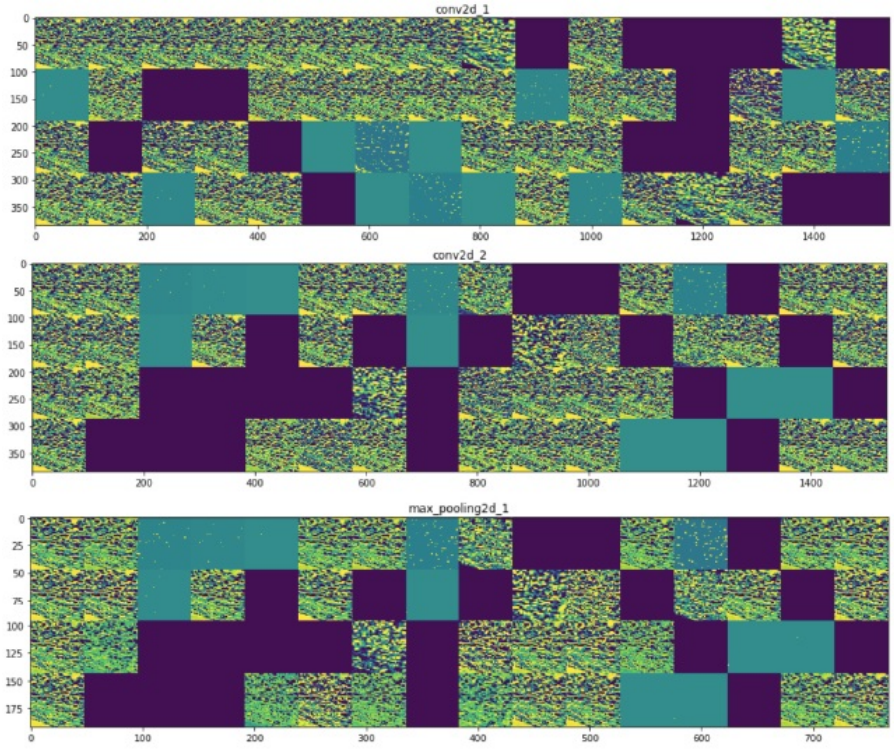


Figure 3.3: Feature extraction in block of convolution.

The training process was continued at the MLP block, in which the backpropagation process was carried out in an iteration. This process aimed to train the model to be closer to the target value or, in other words, to minimize the value of error/loss. As previously mentioned, this architecture involved the three MLP layers, which were two Dense 64 channels with ReLu activation function and one Dense two channels with Sigmoid activation function. Thus, the backpropagation process calculation can use the Sigmoid Cross

Entropy Loss or the so-called Binary Cross-Entropy Loss. This function is a combination of the Cross-Entropy Loss Function with the Sigmoid Activation Function. Because each class's value did not influence the other, this function was called an independent function. Before we see how the function was implemented, let us first look at Cross-Entropy Loss, namely:

$$CE = -\sum_{i=1}^{C} t_i \log(s_i), \tag{3.1}$$

where t_i is the target value and s_i is the score for the i^{th} classes. Because this is binary classification problem with $C = 2$, then cross entropy loss for binary classification problem can be expressed by:

$$CE = -\sum_{i=1}^{2} t_i \log(s_i), \tag{3.2}$$

or this is straightforward to:

$$CE = -t_1 log(s_1) - (1 - t_1) log(1 - s_1). \tag{3.3}$$

As we know s_i is calculated by Sigmoid Activation Function, when we use this loss function, the formula for Cross Entropy Loss for Binary Classification problem can be expressed as [45, 11]:

$$CE = -t_1 log(f(s_1)) - (1 - t_1) log(1 - f(s_1)), \tag{3.4}$$

where the sigmoid activation function represented by:

$$f(s_i) = \frac{1}{1 + e_i^s}. \tag{3.5}$$

The training process of the model produced accuracy and validation values. The accuracy value represents whether or not the architecture is well designed, while the validation value describes how well the architecture's parameters work. The accuracy value itself was calculated by the composition of the amount of predicted data as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}, \tag{3.6}$$

where TP is true positive, that is data in class 0 (no cancer) and precisely predicted as class 0, TN is true negative, that is data in class 1 (cancer) and correctly predicted as class 1. FP is false positive, namely data in class 1, which is incorrectly predicted as class 0, and conversely, FN is false negative as data in class 0, which is mistakenly predicted as class 1.

From the training process of 50 epochs, it can be reported that the architecture seems to be well implemented by achieving an accuracy value of 0.80 on the first epoch and 0.91 on the fiftieth epoch. Likewise, the model's parameters can be appropriately implemented, evidenced by the validation value for the first epoch,

which was 0.83 and at the last epoch, 0.92, as shown in Figure 3.4.

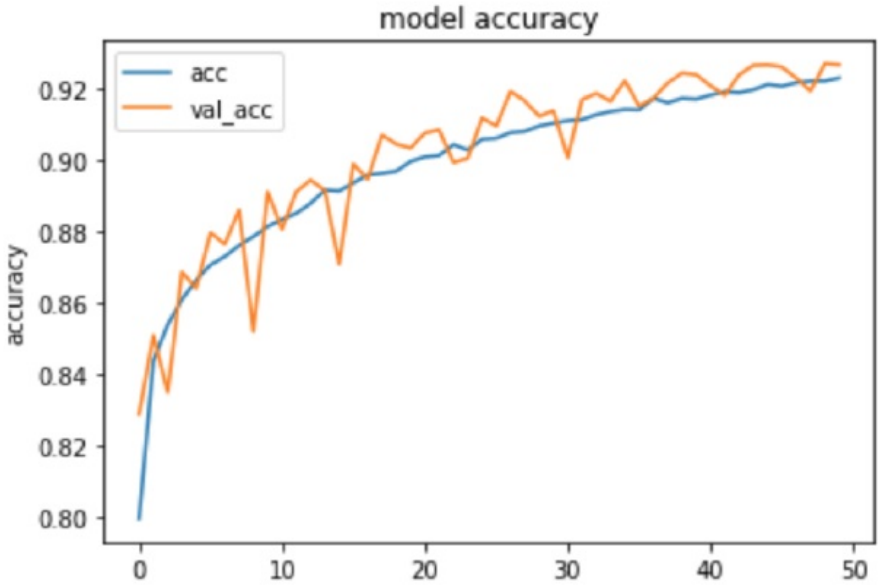


Figure 3.4: Model accuracy.

Likewise, the backpropagation process in the model went well. It is shown in Figure 3.5, where there was a significant and consistent decrease in the loss value during the training and validation. The initial loss value during the training was 0.45, then drops to 0.22 at the fiftieth epoch. Simultaneously, the validation process had an initial loss value of 0.39 and ends with a loss value of 0.19 in the last epoch. The validation value and training accuracy of the model that was not far indicated that the model was well implemented and did not experience overfitting.

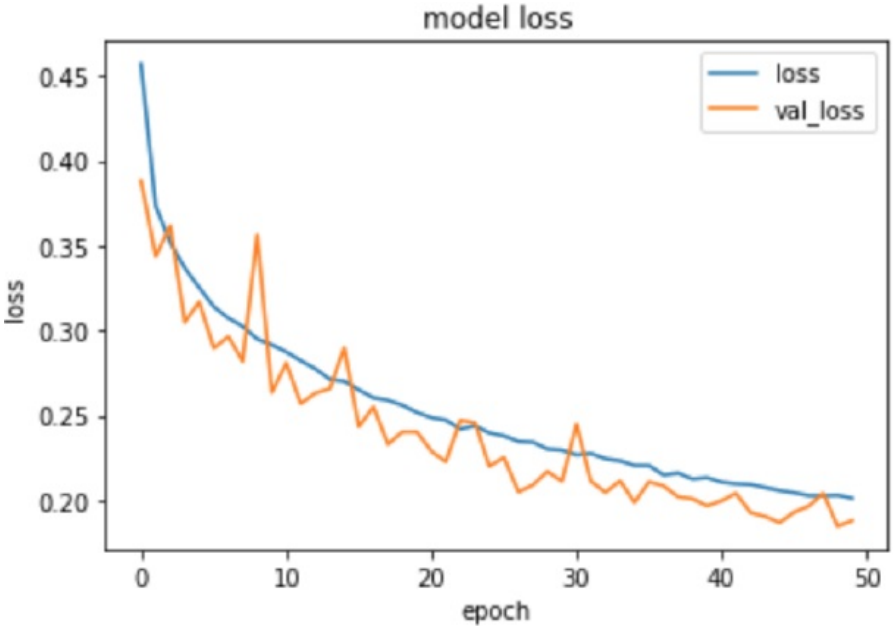


Figure 3.5: Model loss.

3.3. Results

After going through the training process, we tried to use a model in predicting data. In this study, we used 66,007 histopathological images as testing data. From Figure 3.6, we can see the prediction results of the entire data. The Y-axis represents the amount of data, and the X-axis represents the score of each data. Furthermore, it can be explained that a score of 1 stated that the image contained cancer cells and vice versa, a score of 0 did not contain cancer cells. For more detail, approximately 29,000 images were

predicted by the model with a high probability of not containing cancer cells, whereas approximately 19,000 images were predicted with a high probability of containing cancer cells. It can also be seen that the proportion of the predicted data gets smaller when it reached the value 0.5, where this value represents the model's hesitation when predicting the data.

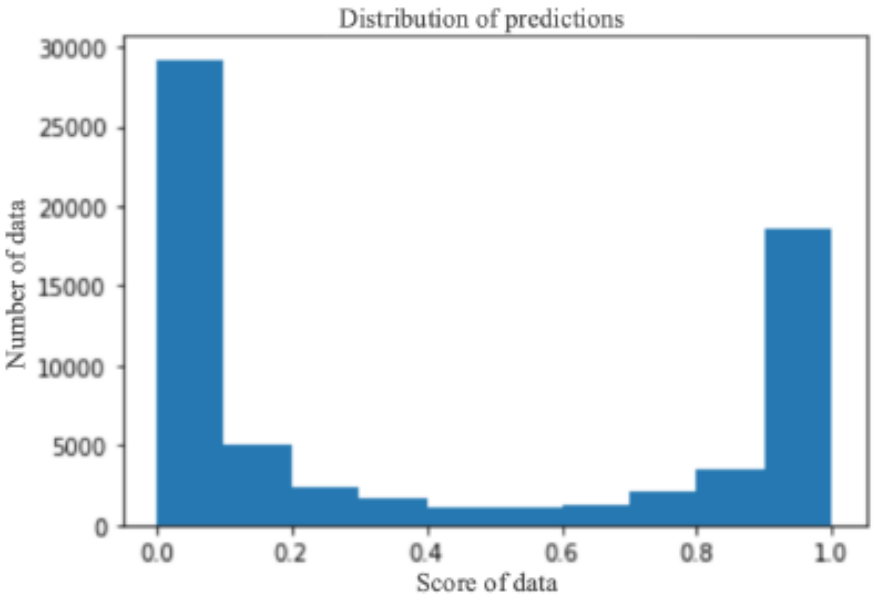


Figure 3.6: Distribution of predictions.

Next, we presented a confusion matrix table that describes the composition of the two classes' prediction results. Of the total images, 39,310 images did not contain cancer cells, and 26,697 were images indicated to have cancer cells. As much as 37,058 normal

images were correctly predicted as normal images, while 2252 normal images were incorrectly predicted as images containing cancer cells. On the other hand, 24,207 images indicated to have cancer cells have correctly predicted, and only 2490 images from this class were incorrectly predicted as normal images as described in Table 3.1.

Table 3.1: Confusion matrix.

x	class no cancer (0)	class cancer (1)
predicted as no cancer (0)	37058	2490
predicted as cancer (1)	2252	24207
support	39310	26697

We also calculated the Area Under Curve (AUC) from the successfully classified data, by preliminary calculated the True Positive Rate value and the False Positive Rate value. Then the two values were plotted in a Receiver Operating Characteristic Curve as shown in Figure 3.7. From this figure, it can be understood that the maximum TPR and FPR value was 1, which represented the probability value of data in a class. The greater the TPR value and the smaller the FPR value, the greater the AUC value obtained.

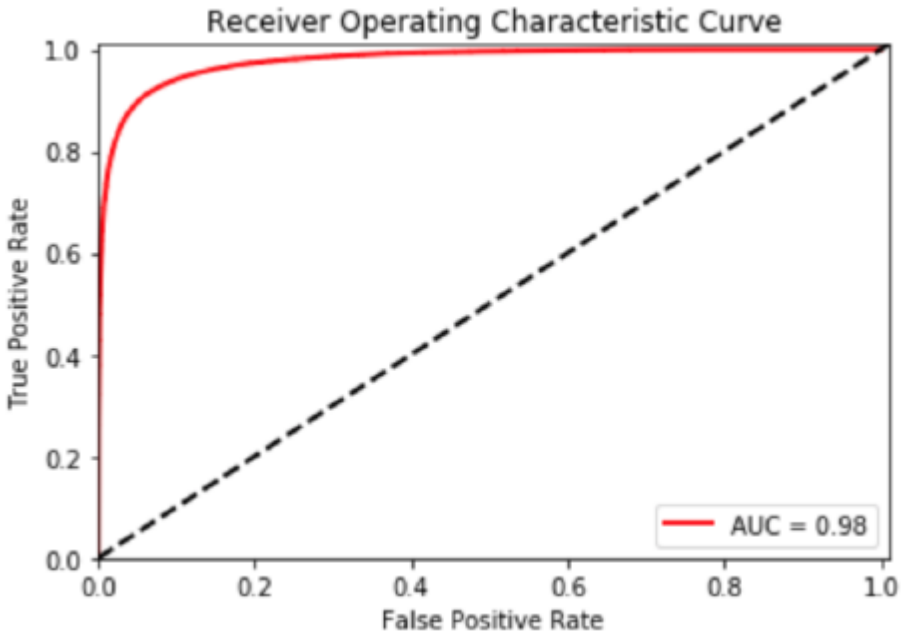


Figure 3.7: ROC and AUC

3.4. Discussion

From the above research results, we can conclude that the Neural Network model with convolutional architecture can be used as a solution to the binary classification problem in predicting cancer on histopathological images. The proposed CNN was proven to study histopathological image features that can distinguish between images containing cancer cells and non-cancer cells. The proposed model also demonstrated excellent performance in classifying histopathological images by producing no more than 1000

images with probability values 0.4 to 0.6. Furthermore, only about 5000 images have a probability value of 0.2 to 0.8. The accuracy value confirmed this result on the confusion matrix, which showed the percentage of errors in the normal image class reached 0.6%, and the image contained cancer cells up to 10%.

4. Ensemble CNN as a comparison for single CNN performance

In this study, we designed several single networks and then combined them into an ensemble model [27]. The developed single network referred to several pre-trained models, which means that even though we were designing the network from scratch, we relied on pre-trained modules that were already there. The pre-trained modules that we used including the VGG module, the Inception module, and the ResNet module.

4.1. Related work

This research process began with mapping the similar works that have been completed. These works were the utilization of CNN to solve classification problems on similar dataset. The first work was using the Dense Block and the Transition Block to design the P4M-DenseNet. This model achieved an accuracy value of 89.8% for predicting histopathological images [55]. In the second work, the GoogleLeNet architecture was used, which was then trained in two different approaches. The first method was training from scratch, and the second was fine-tuning. The results showed that fine-tuning n the training process produced the best accuracy value with 84.3% [58]. The third study was utilized the PatchCam

dataset to demonstrate the designed ensemble model [22]. The ensemble model was trained using the transfer learning method by utilizing three pre-trained models, namely VGG19, MobileNet and DenseNet. The model achieved satisfactory results of a 94.64 accuracy score.

4.2. Data preprocessing

As explained in the previous chapter, an augmentation process was first carried out on the dataset before entering the training process. The augmentation process includes rotation, shifting, shearing, zoom and flipping as shown in Table 4.1.

Rotation	45°
Shifting	0.2
Shearing	0.2
Zoom	0.2
Flipping	Horizontal

Table 4.1: Augmentation process.

4.3. Network architecture

As mentioned earlier, we used the VGG, Inception, and ResNet modules to design the ensemble model. To avoid very complex models, we designed a simpler architecture than the existing pre-

trained models. Thus, the first model was the LT-VGG which adopted from the VGG module. This architecture was a stack of thirteen layers consisting of ten Convolution layers and three Fully Connected layers. A Max Pooling layer was inserted in every two convolution layers to parse its dimensions. Then the Flatten layer changed the dimensions of the features to be passed on to the MLP. Within the MLP itself, there were three fully connected layers, each with two 64-neurons and a layer of two neurons. All activation functions in this architecture used the ReLu activation function except for the final layer used the Softmax activation function according to the dataset's problem.

The second model was the LT-Inception which adopted from the Inception module. The architecture in this model consisted of twelve convolution layers, which were divided into two levels. The architecture at both levels was identical, consisting of six convolution layers and ends with the Max Pooling layer. Afterward, the layers at each level were assembled. To reduce the number of dimensions of features, the Average Pooling layer was used. Finally, the feature was transmitted in an MLP consisting of two Fully Connected 64-neurons and a Fully Connected two neurons. The LT-Inception model used the ReLu Activation Function on almost all networks except for the last layer; Softmax two classes.

LT-ResNet was the last model we used in the designed ensem-

ble. Using the ResNet module, we designed a total of 24 convolution layers. In detail, we installed eighteen convolution layers by inserting a residual layer on each of the three convolution layers. Like the previous model, LT-ResNet also used the Average Pooling layer and the Flatten layer before entering MLP. Furthermore, ended with the MLP consisted of three fully connected layers as in the two previous models.

The ensemble method is one of the popular techniques to improve CNN's accuracy, as described in [20, 46, 14]. In this work, we compared each model's performance with the performance of all three models simultaneously. The ensemble method used was a voting system. Figure 4.1 is the design of our proposed model of ensemble.

4.4. Training process

The histopathological image training process was initiated in every single tissue. Each network exercised separately did not interfere with one another. Because the training process did not use the transfer learning method, initial weights were needed in the training process. Realizing that the three single networks' architecture has more than eight layers of convolution with non-linear activation, Normal Distribution [18] was used as initial weights.

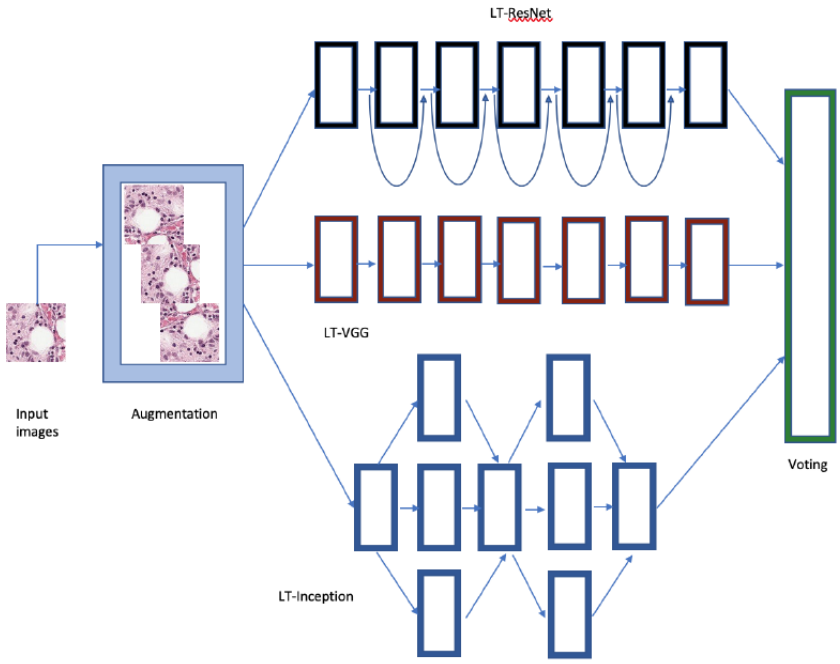


Figure 4.1: Architecture of the ensemble model.

The use of the optimizer was also part of the hyperparameter of the training process so that the Adam[24] optimizer was used with a learning rate of 1e-4 and reduced by 1e-6 for each subsequent epoch. To reduce the computational load, the batch size was implemented in the training process. Accuracy, Precision and Recall values were some of the parameters used in determining whether or not the model was good. To calculate the value of Accuracy we were using equation (3.6) and for these two parameters, we referred to the following formula:

$$Precision = \frac{TP}{TP + FP}, \tag{4.1}$$

$$Recall = \frac{TP}{TP + FN}, \text{ where} \tag{4.2}$$

TP stands for true positive, where an image containing cancer cells was accurately predicted as an image containing cancer cells. Meanwhile, TN is the true negative, which was a normal image and was predicted accurately as a normal image. FP stands for false positive, a normal image but was mistakenly predicted as an image containing cancer cells. In contrast, FN stands for false negative, representing an image containing cancer cells but was incorrectly predicted as a normal image. Apart from the parameters mentioned above, another parameter that affected was the loss value. The smaller the model's loss value, the closer the model is to the expected target. In this study, the Softmax activation function was used at the end of the network. Thus, to calculate the loss value, cross-entropy was used for the Softmax loss function. Softmax function $f(s)_i$ is represented by:

$$f(s)_i = \frac{e^{s_i}}{\sum_{c=1}^{K} e^{s_c}}. \tag{4.3}$$

Refers to [45, 12], Softmax function $f(s) : \mathbb{R}^K \to \mathbb{R}^K$ is a vector function in the range $[0, 1]$, where K is the number of classes. This

function is obtained by calculating the exponential number to the power of s_i, where s_i refers to the score s from class i. Hereafter, numerator divided by the sum of the constant e to the power of all score in number of classes. By having equation (4.3) then we have the Softmax loss function:

$$CE = -\sum_{i}^{K} t_i log(f(s)_i). \tag{4.4}$$

Equation 4.4 explains that cross-entropy CE is the sum of ground truth t_i logarithm the CNN score of each class that represents by $f(s)_i$.

The ensemble network's training process included a separate training process for every single network. Thus, every single network provided a different representation of the trained image. Alternatively, in other words, the model provided three different outputs for each given input. For this reason, the model required a system that combines the three outputs into one output as a mutual agreement of the three models. The method used in this process was a voting system. The voting system itself referred to two things. The first was that every single network has equal weight in voting. In other words, every single network has the same ability to influence the final result. This voting system was called Majority Voting. The second voting system was assigning different weights

to every single network. The weight given corresponded to the capability of a single network when trained separately. Thus, a single network that has a good accuracy value will affect the final result more significantly with other single networks. This voting system was called Weighted Vote. Voting system itself refers to [13, 37], if we have multiple scores x_1, x_2, ..., x_n, with corresponding weights w_1, w_2, ..., w_n, then the weighted mean can be calculated through

$$\bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}. \tag{4.5}$$

4.5. Results

After passing through the training process of 50 epochs, the results showed that the decrease in the loss value of LT-ResNet and LT-Inception was relatively stable. Meanwhile, LT-VGG showed fluctuating movements after the twentieth epoch. The three models were not much different to achieve the loss value, namely, 0.13 for LT-ResNet, 0.19 for LT-Inception, and 0.26 for LT-VGG. Thus, from the training results, it was found that the LT-ResNet model has a better loss value than the other two models. The loss score achievement of the three single models is shown in Figures 4.2, 4.3 and, 4.4

The training results can also be reported that the three single

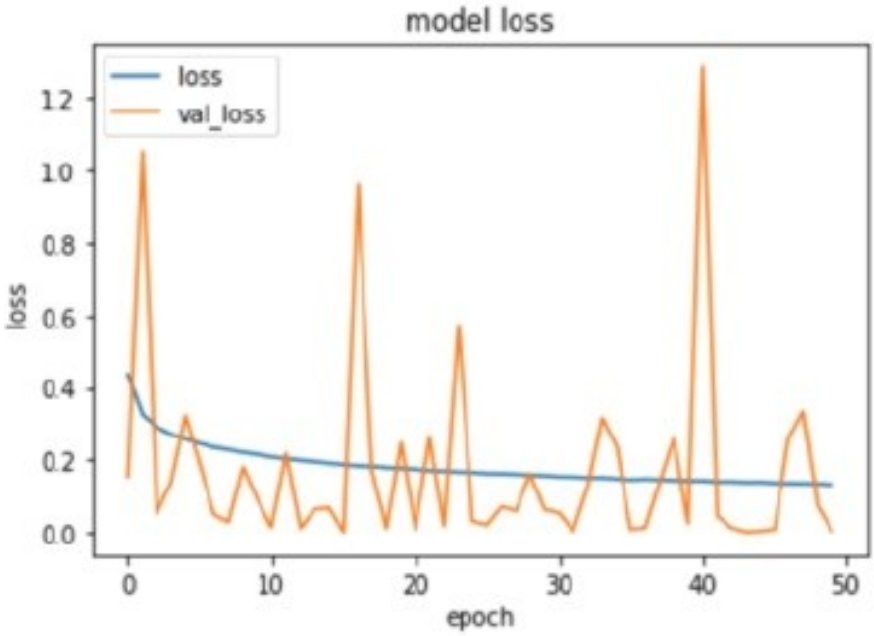


Figure 4.2: Loss of LT-ResNet.

models did not experience overfitting, proven by the accuracy and validation values of the training process, which were not distant. In addition, the three single models showed a significant increase in accuracy and validation values from the first epoch to the last epoch. From Figures 4.5, 4.6 and, 4.7, it can be reported that the LT-ResNet model achieved accuracy and validation values of up to 0.95, the LT-Inception model achieved accuracy and validation values of up to 0.93, and LT-VGG reached accuracy and validation values of up to 0.95. Furthermore, the precision and recall values of every single network in each class were calculated. The results

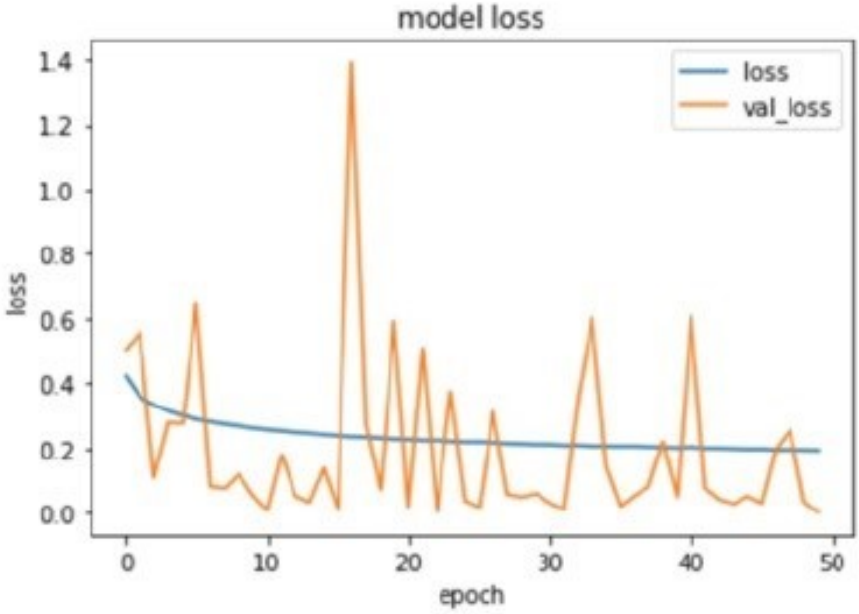


Figure 4.3: Loss of LT-Inception.

obtained was the LT ResNet achieved a precision score of 0.95 with a recall of 0.97 for normal category images and a precision score of 0.95, and a recall of 0.92 for images containing cancer cells. The LT-Inception model achieved a precision value of 0.92 and a recall of 0.96 for normal images, while for images containing cancer cells, 0.93 and 0.89, respectively, for the precision and recall scores. The VGG model showed excellent results with precision and recall values of 0.95 and 0.97, respectively for normal images, while for images containing cancer cells, the precision and recall values were 0.96 and 0.92, respectively. From the precision and recall

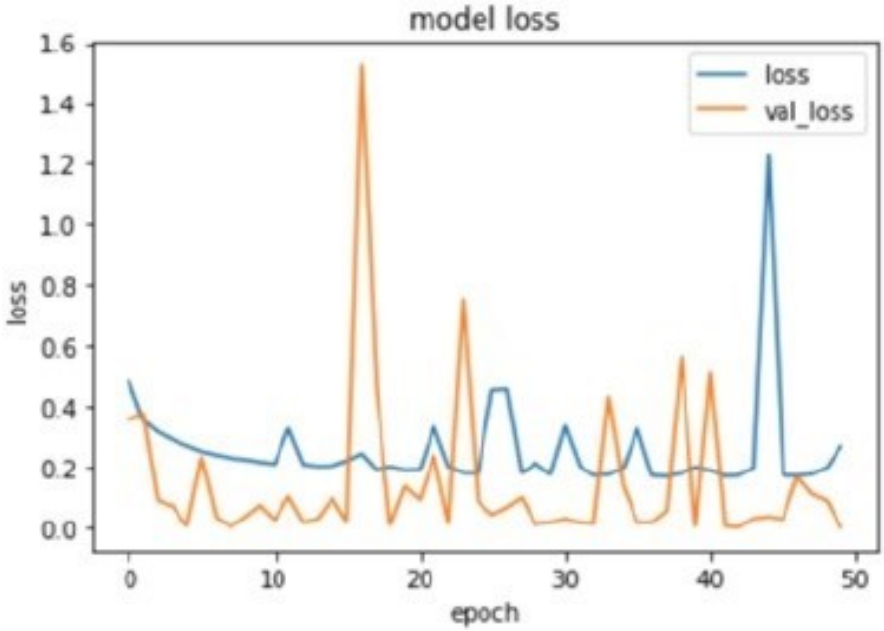


Figure 4.4: Loss of LT-VGG.

calculations, it was clear that the LT-ResNet and LT-VGG models have slightly better capabilities than the LT-Inception model.

From the training results using the majority voting method, it was found that the ensemble model can correct the accuracy value of a single model up to 0.96, confirmed by the fact that the value of the precision and recall of the ensemble model was also better than the single network model. For normal images, the ensemble model achieved a precision value of 0.95 and a recall value of 0.98, while for images containing cancer cells, the ensemble model achieved precision and recall values of 0.96 and 0.93, respectively. After

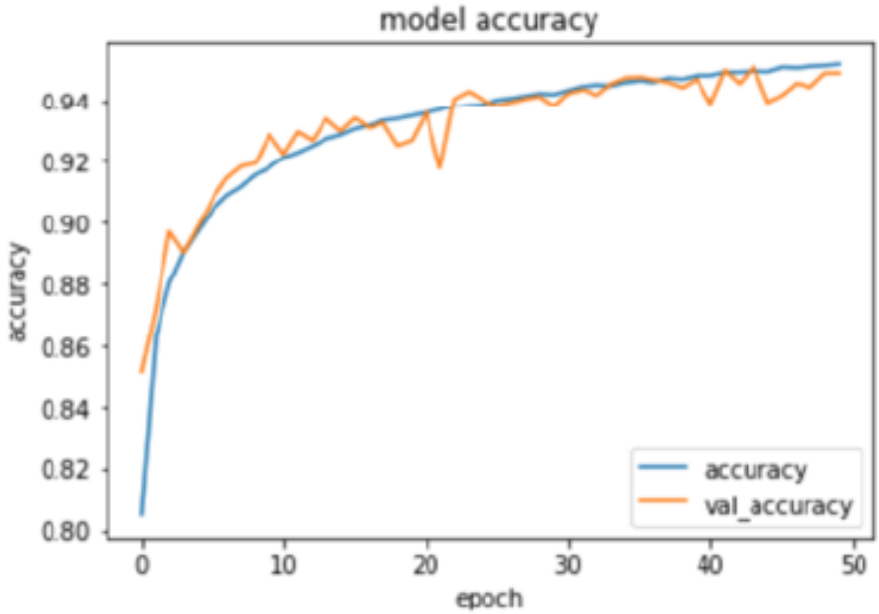


Figure 4.5: Accuracy of LT-ResNet.

obtaining the training results for each single network, it was decided that the LT-ResNet and LT-VGG models each received a weight of 35%, while LT-Inception received a weight of 30% in the weighted voting process. The results of weighted voting also reported the same result as the majority vote method, namely an accuracy value of up to 0.96 as shown in Table 4.2. This confirmed that in our experiment, every single network received a suitable weight for the voting process.

To see more detail the model's capabilities, we calculated the amount of the data either correctly or incorrectly predicted in the

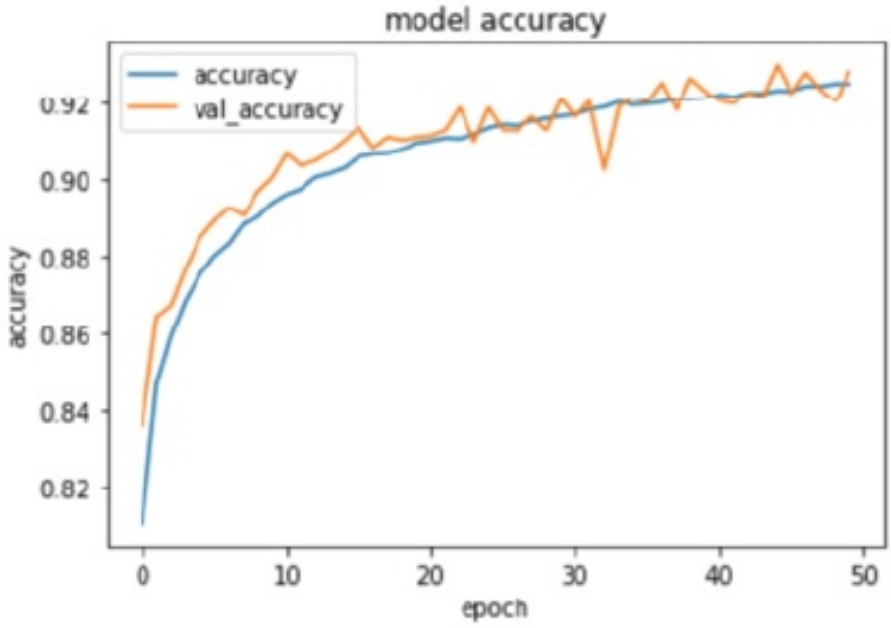


Figure 4.6: Accuracy of LT-Inception.

x	LT-ResNet	LT-Inception	LT-VGG	MV	WMV
Pre 0	0.95	0.92	0.95	0.95	0.95
Rec 0	0.97	0.96	0.97	0.98	0.98
Pre 1	0.95	0.93	0.96	0.96	0.96
Rec 1	0.92	0.89	0.92	0.93	0.93
Acc	0.95	0.93	0.95	0.96	0.96

Table 4.2: Precision, Recall and Accuracy of the investigated models.

two classes. LT-ResNet was able to predict precisely 38,043 cancer images with 1653 images of errors. Meanwhile, the normal image that was predicted correctly was 24677 with an error rate of 2020.

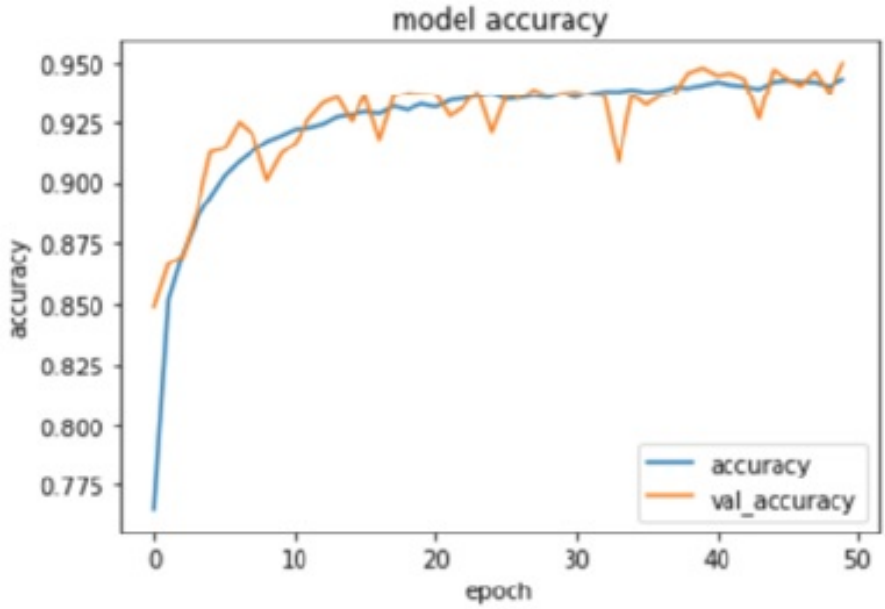


Figure 4.7: Accuracy of LT-VGG.

Furthermore, LT-Inception was able to accurately predict 37.657 images containing cancer cells with an error of 1653 images. As for normal images, 23,634 images were predicted correctly with an error of 3063 images. LT-VGG correctly classified 38,186 cancer images and 1124 images were misclassified. In the normal image class, 24,504 images were correctly classified, and 2193 images that were misclassified. As previously explained, the ensemble model performed better than all single models. The results achieved by the majority voting system was 38,392 cancer images were successfully predicted with an error of 918 images. More than that,

24,773 normal images were predicted correctly with 1924 image errors. The majority voting results were confirmed by the results of the weighted majority vote, with 38,397 images predicted as images containing cancer cells and 913 images were misclassified. As much as 24,787 images were correctly predicted to be normal, with the error reaching 1910 images. Table 4.3 shows the comparison of the amount of data predicted correctly and incorrectly by all models.

x	LT-ResNet	LT-Inception	LT-VGG	MV	WMV
TP	38043	37657	38186	38392	**38397**
TN	24677	23634	24504	24773	**24787**
FP	2020	3063	2193	1924	**1910**
FN	1265	1653	1124	918	**913**

Table 4.3: Confusion matrix of the investigated models.

4.6. Discussion

This work has shown the comparison of the results obtained from several similar jobs, which resolved classification problems on histopathological images. Table 4.4 shows a comparison of the accuracy scores of the methods we recommend with other methods. From this table it can be concluded that the ensemble method can improve the accuracy of the model. This can be seen from the two ensemble model that we suggested, compared to these works [55, 58]. In addition, developing a single model from scratch can help

the model to achieve optimal accuracy values without reducing the model's ability in classification work. Thus, the combination of these two methods yielded the best accuracy value in our experimental case. This was evidenced by comparing the results shown by the proposed methods with this work [22]. Another conclusion was that manual weighting on a single network for weighted voting did not show a significant impact, although there was a slight improvement in the predicted data.

Method	Architecture	Accuracy
Veeling et al.	P4M-DenseNet	89.8%
Xia et al.	GoogleLeNet fine-tuned	84.3%
Kassani et al.	Ensemble	94.64%
Proposed method 1	LT-ResNet	95%
Proposed method 2	LT-Inception	93%
Proposed method 3	LT-VGG	95%
Proposed method 4	Ensemble MV	96%
Proposed method 5	Ensemble WMV	96%

Table 4.4: Comparison results.

Referring to the references involved in this research, several methods can be used to improve the model's performance. One of them was the Grid method to determine the correct training parameters for the model. The matter to consider was the machine's ability; it must be suitable for achieving optimal results. We also considered including the voting process as part of the training parameters. Thus, the weighting will not be given arbitrarily, but

the training process itself will determine which single network has more influence on the ensemble model.

5. Comparison of single, ensemble majority voting and, interconnected CNN performance in the medical images classification task

In the previous chapter, it was shown that the ensemble CNN with the voting method achieved better accuracy values than the Single CNN. It was also found that the weighting on the system voting did not significantly impact the accuracy scores of the ensemble model. Follow up on the findings that there was no significant impact of weighting for every single network, then the next experiment focused on making the weighting process part of the training parameters. In other words, the training process itself will determine which single network influences the ensemble model more when making decisions. This chapter reports the results of investigating the efficiency of interconnected CNNs in classifying the medical images. For objectivity purposes, three different datasets were used in this experiment.

5.1. Datasets

As previously mentioned, the model developed in this work will then be trained on three different datasets. All three datasets

were open data republished by Kaggle. One of them published in the form of a competition, while others in the form of a dataset. All three datasets were medical images resulting from the process of digitizing the human body. The first dataset was the chest X-ray[1] dataset, representing the least amount compared to other datasets. A total of 5216 radiological images of human lungs were grouped into two not proportionally distributed classes. Namely, 3875 images containing pneumonia and 1341 images not containing pneumonia. All images were labeled by a doctor and verified by an expert. The sample image on the first dataset is shown in Figure 5.1. The second dataset was the Malaria[2] dataset, representing the medium dataset, which amounted to 27,560 images proportionally distributed in both classes. The Open Knowledge Foundation[3] owns the malaria dataset was later republished in the form of the Kaggle dataset. The image was the digitization of the Thin Blood Smear using a microscope application integrated into the android smartphone. In the end, the dataset labeling process was carried out by experts with two categories, namely, parasitized for images containing malaria and normal for images that did not contain malaria. Figure 5.2 shows an example of the Malaria dataset. The third dataset was the dataset that has been used in the previous

[1] https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia
[2] https://www.kaggle.com/miracle9to9/files1
[3] https://opendatacommons.org/licenses/by/1-0/index.html

work, namely the PatchCam[4] dataset republished by Kaggle in the form of a competition. As previously explained, these data sets were 220,025 small pathology images, categorized in two classes: cancerous and non-cancerous, as shown in Figure 5.3.

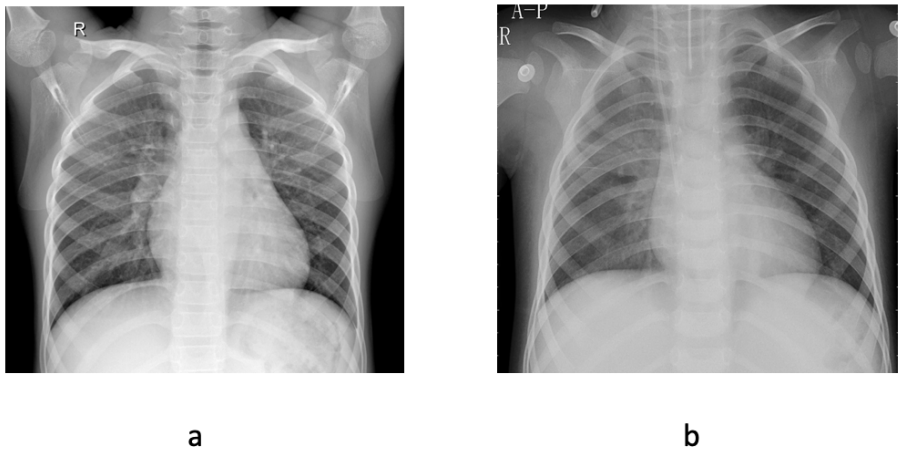


Figure 5.1: X-ray dataset, (a) Normal and (b) bacterial or viral pneumonia

5.2. Network architecture

The interconnected CNN architecture is a combination of several CNNs. In this work, the three single networks were connected, which then trained together in a more extensive network to have a single decision [28]. The purpose of connecting the three single

[4] https://www.kaggle.com/c/histopathologic-cancer-detection

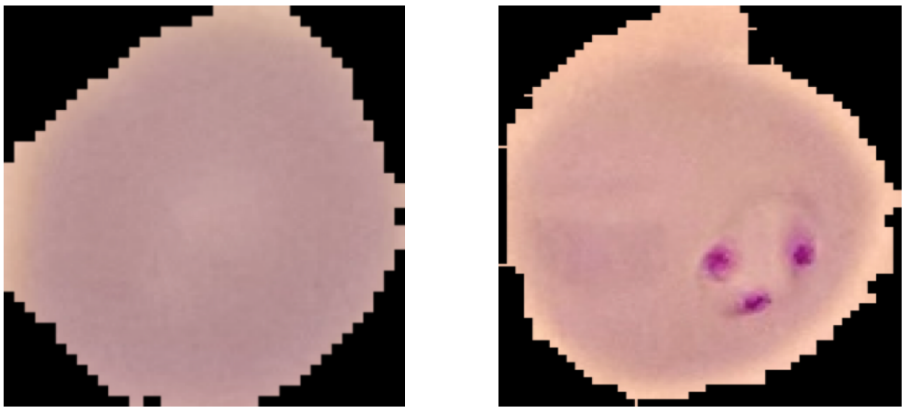


Figure 5.2: Malaria dataset, (a) Normal and (b) parasitized

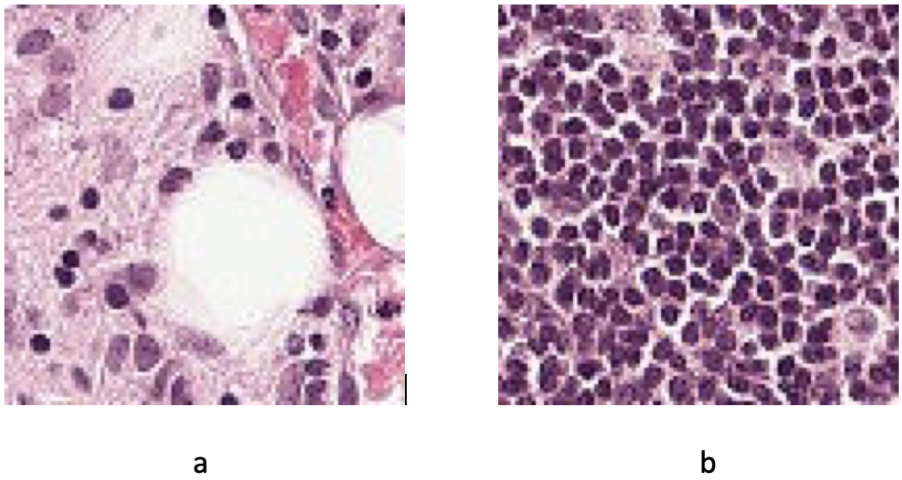


Figure 5.3: PatchCam dataset, (a) Cancerous and (b) non-cancerous

networks was that even though the three single networks work independently, the results of training for every single network will de-

termine how much this network affects the interconnected model. Thus, this training process will give the proper weight to every single network to influence the interconnected model in decision making.

In this work, we used three single networks, which were the pre-trained model. The three pre-trained models were VGG19 which consisted of 16-layer convolution; InceptionV3, which utilized 48-layer convolution; and MobileNet, which installed 18-layer convolution. After going through the convolution layer, the dimensions of the feature map need to be adjusted as required in the Multi-layer Perceptron (MLP). For this reason, before the MLP section, a Flatten layer was installed to change the dimensions of the features. The MLP of each network was replaced with three fully connected layers to adjust to the problems in the dataset. Two of the fully connected layers used ReLu activation functions, while the other used the two-class Softmax function. In the next stage, the three single networks connected become one ensemble network. For this reason, the Concatenation layer was used to unite the three single networks into an Interconnected CNN network. After the three single networks were united, MLP consisted of two Fully Connected layers with a ReLu activation function, one final layer with two classes of Softmax was re-installed. The design architecture of the Interconnected model is shown in Figure 5.4.

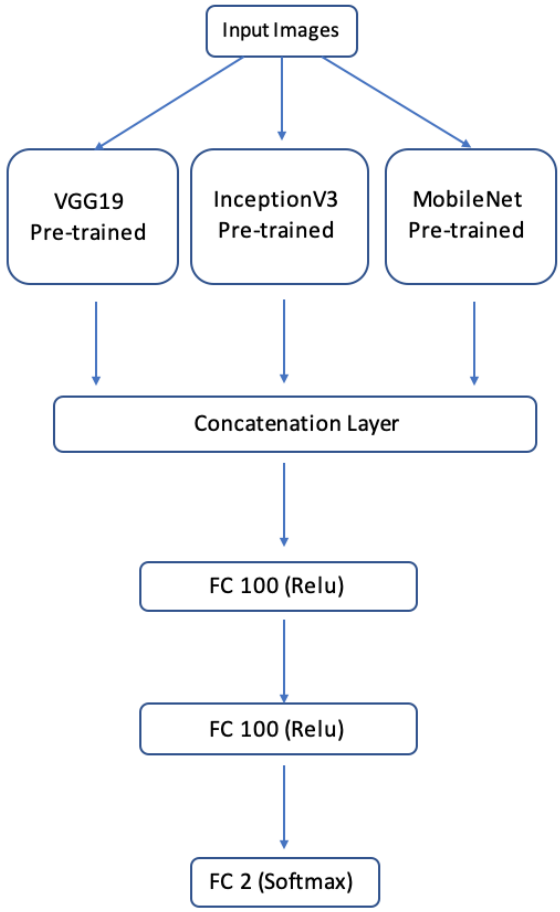


Figure 5.4: Architecture of the interconnected Model

5.3. Training process

Like previous works, this experiment was also through an augmentation process in Table 4.1 to ensure the right amount of data needed in the training process. Stages such as rotation, shifting,

shearing, zoom, and flipping were techniques used in the augmentation process. Several aspects have been standardized for simplifying the training process for four networks on three datasets. The first aspect was the input size set to 100x100 pixels, as well as the batch size set at 16 and the number of epochs at 50 for each training process. Before starting the training process, 10% of the total dataset has been separated into test sets. Subsequently, the dataset was trained with a proportion of 70% for the training set and 30% as a validation set.

In our work, the emphasis was on having a training process that was carried out simultaneously in a series of interconnections. Although each sub-network has authority in the training process, the training process was an unseparated integral part, which caused each sub-network's weights to be determined by the training process itself and not by the user. The sub-network will automatically impact the overall interconnected model if the subnetwork has a better performance than others. Contrariwise, a sub-network produced unsatisfactory performance will lead to less weight in the final decision process. In [14], for the three sub-networks, the initial weights were calculated to be equal. Nonetheless, the weights of the interconnected models in our model were calculated by the training phase that each model had gone through. At neither the beginning nor the end of the decision-making phase, the weighting

process of each sub-network was interfered with.

In order to consider the adequacy of the amount of data and computation costs, the Transfer learning training method was used in this experiment. Transfer learning refers to the condition where what has been learned in one dataset is exploited to improve generalization in another dataset [12]. Furthermore, Transfer learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned [54]. By utilizing Transfer Learning, unneeded layers can be frozen during the training process. The Freezing layers technique was used to deactivate the layers that were not required and prevent them from consuming computational resources. As explained in [7], this technique can disable the unneeded layers in the training process but did not affect the model's performance. The freezing layers process included several layers that were deactivated during the training process so that the update weights process did not occur when information passes through the layers.

One of the goals of the training process was to get the least possible loss. Thus the training process required a function that can minimize the value of the loss. In this case, the Soft-Max function was implemented in the Loss function or better known as the Soft-Max Loss function. How the Soft-Max function was implemented in the Loss function has been explained in the previous

chapter. From the explanation above, we can understand that in order to get a minimum loss value, there was an iterative process in calculating the loss value. This iteration process used a propagation technique that involves an optimization technique known as the Adam optimizer. In the optimization process itself, the initial rate was determined, set to 1e-4 and 1e-6 decay for each subsequent epoch. In the end, the training process will produce several parameters, namely Accuracy, Precision, and Recall. These three parameters were the representation of whether or not a model is adequate. The values of the three parameters were obtained from True Positive, True Negative, False Positive, and False Negative values. How the values of these parameters were calculated has been explained in the previous chapter.

5.4. Results

The first experiment result was the training and validation of every single network that was trained separately. There were several goals to be achieved by training the three models separately. The first was that the three models need to prove their respective performance when trained on three different datasets. In other words, it is necessary to consider the ability of the model individually before being assigned as a member of the interconnected

model. The second reason was that the voting-based ensemble model, which will be used as a performance comparison of the interconnected model, needs to train the three models individually.

Using the chest X-ray dataset, the three single networks could be implemented properly from the training process. The accuracy and validation values of the three models were not significantly different. This was also evidenced by the yellow and blue lines in Figure 5.5, which overlapped and even intersect. This condition proved that the parameters used in the model were suitable so that the model did not overfit. Each model achieved a validation value of 0.91 for VGG19, 0.89 for InceptionV3 and 0.93 for MobileNet. The training process using the Malaria dataset also reported satisfactory results, as shown in Figure 5.6. Even though the results shown by MobileNet have a gap between the accuracy and validation values, the movement of the yellow line that increases every epoch can be assumed this model was still acceptable. The respective validation values achieved by a single network are 0.90 for VGG19 and 0.80 for InceptionV3 and MobileNet. A slight difference was found when the three models were trained on the PatchCam dataset where MobileNet showed a slight overfit condition as shown in Figure 5.7. However, the other two models showed good performance. The validation values achieved by the three models when trained with the PatchCam dataset were 0.86 for VGG19, and

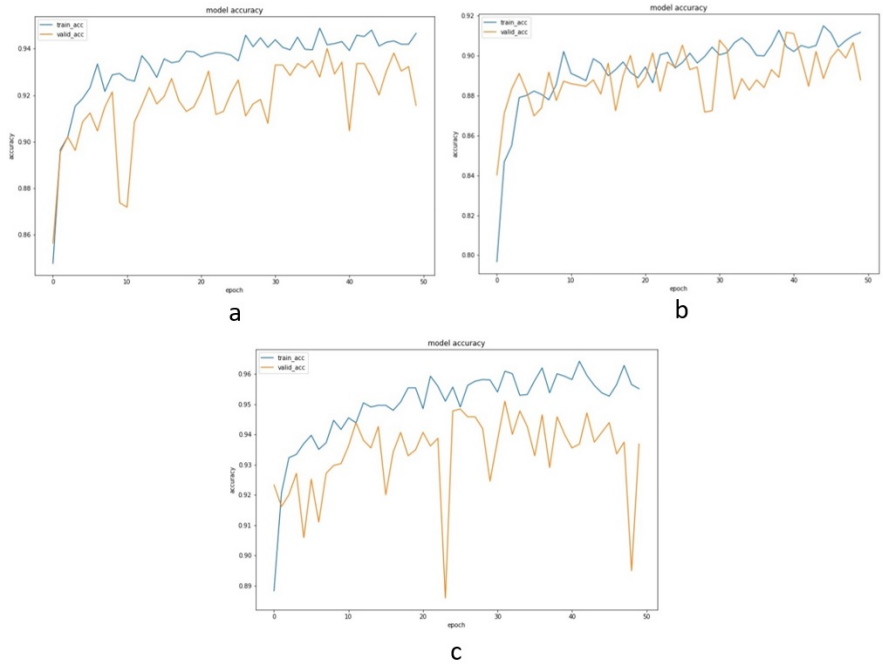


Figure 5.5: Training and validation accuracy for X-ray dataset: a) VGG19, b) InceptionV3 and, c) MobileNet

0.70 for InceptionV3 and MobileNet models. The second experiment was to train all three single networks together. If the three networks were trained separately in the first experiment, then in this experiment, the three networks were trained simultaneously in a connected network. In other words, in this process, all parameters were trained together. From the training results, it was found that the accuracy value of the interconnected models can be compared with each single model. The comparison of the accuracy values of a single network and interconnected models is shown in Table 5.1.

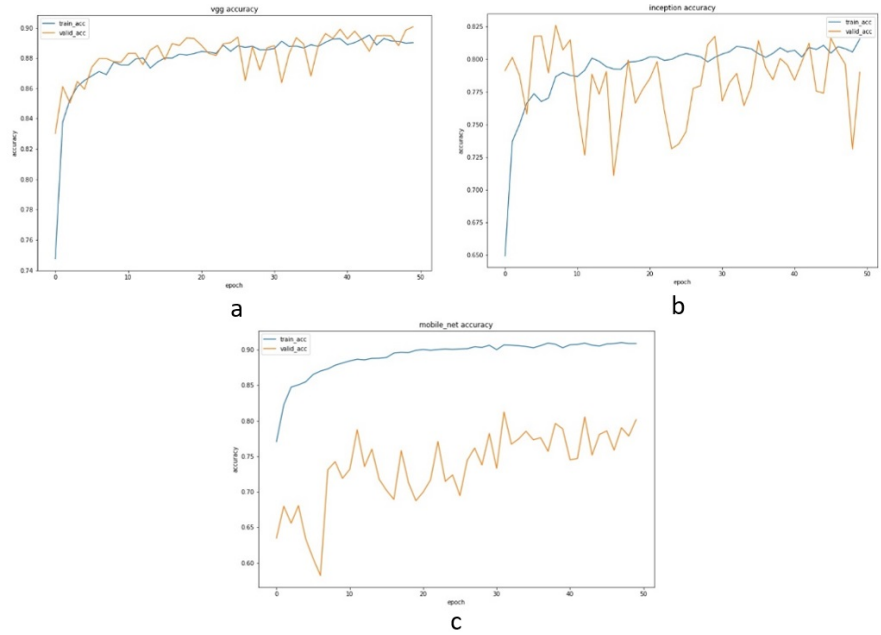


Figure 5.6: Training and validation accuracy for malaria dataset: a) VGG19, b) InceptionV3 and, c) MobileNet

Dataset	VGG19	InceptionV3	MobileNet	Interconnected
chest X-ray	0.91	0.89	0.93	0.93
Malaria	0.90	0.80	0.80	0.90
PatchCam	0.86	0.70	0.70	0.86

Table 5.1: Validation accuracy of all models for three datasets.

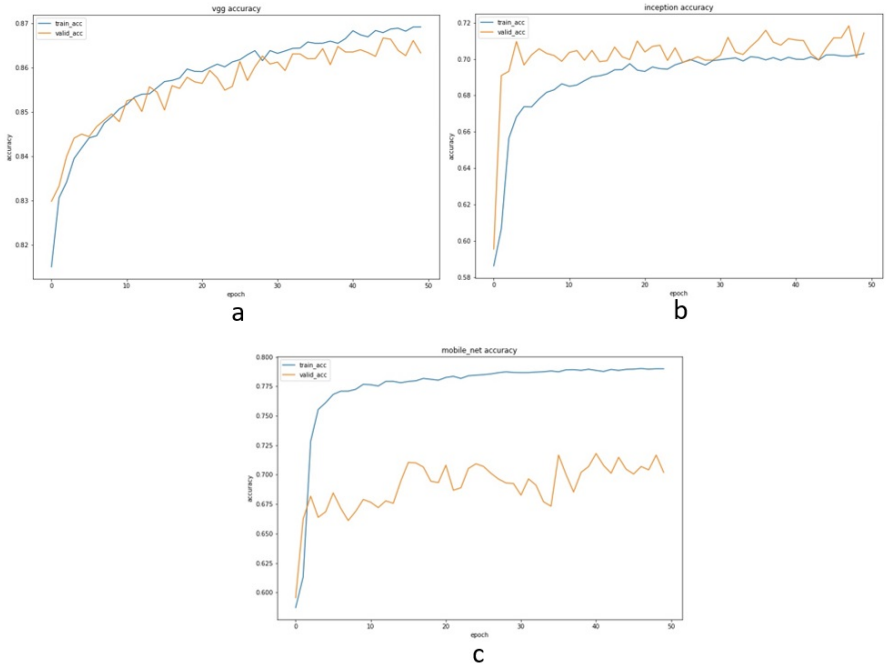


Figure 5.7: Training and validation accuracy for cancerous dataset: a) VGG19, b) InceptionV3 and, c) MobileNet

5.5. Visualization of training process

This section describes the stages in the training process by providing visualization of several images from the dataset. From the chest X-ray dataset, the model identified pneumonia by studying the integrity of the images of human lungs as a result of the radiological process. Figure 5.8a is the input images given to the model. The training process then produced features such as those in Figures 5.8b to 5.8d. By studying these features, the model can

then give the output as shown in Figure 5.8e that most of the lungs still appeared healthy. Thus the image was categorized as normal lungs. For the malaria dataset, Figure 5.9a is the input image given to the model. Based on the input, the model extracted the features as shown in Figures 5.9b to 5.9d. From these features, it can be seen that the model can detect other objects that were not supposed to exist in human Blood Smear Images. Based on the features that indicated an unusual object in the image above, Figure 5.9e is the output given by the model in the form of a heat map which showed a tendency in the image above Blood Smear Images of malaria patients. The training process on the PatchCam dataset aimed to detect the presence of cancer cells in human histopathology images. Figure 5.10a is the input image input of the model. As before, the model extracted features that were considered unique concerning cancer cells. Some examples of images that represent the feature extraction process are Figures 5.10b to 5.10d. After examining the features, the model can conclude that the image above has cancer cells, as shown in Figure 5.10e. The heatmap in Figure 5.10e represents cancer cells with a lighter color than normal cells which were represented in a dimmer color.

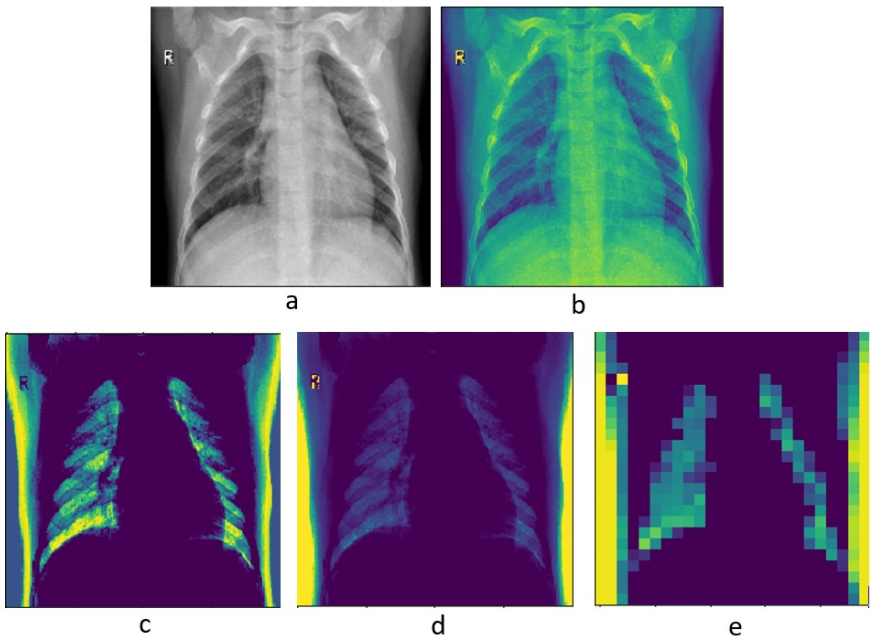


Figure 5.8: a) Input, b)-d) Extracted Features, e) Heatmap

5.6. Predicted results

To see the model's ability to predict new images, a dataset that has never been used before in the training process was used, namely 10% of the images from each previously separated dataset. The predicting the chest X-ray dataset showed that the accuracy values achieved were 0.91, 0.84 and 0.91, respectively, for the VGG19, Inceptionv3, and MobileNet models. In this case, although the Interconnected model cannot exceed the accuracy value of the Majority Voting, the results of the two models can still be compared. In more

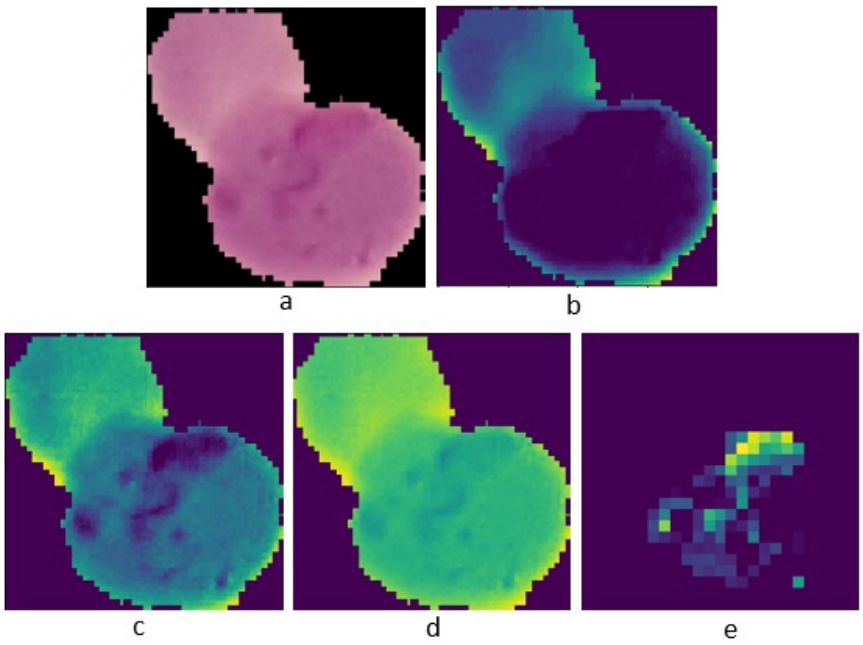


Figure 5.9: a) Input, b)-d) Extracted features, e) Heatmap

detail, Table 5.2 provides a breakdown of the number of images that were predicted right or wrong in the two classes. From the values of Precision and Recall in Table 5.2, it was found that the ability of the Interconnected Model was slightly better than the majority voting. A total of 383 images indicated pneumonia could be predicted correctly and 7 images were mispredicted. Meanwhile, Majority Voting predicted images that indicate pneumonia as many as 370 images with 20 images were mispredicted. The number above was equivalent to the recall value of 0.98 for the Interconnected Model and 0.85 for the Majority Voting. Even so, the Precision values for

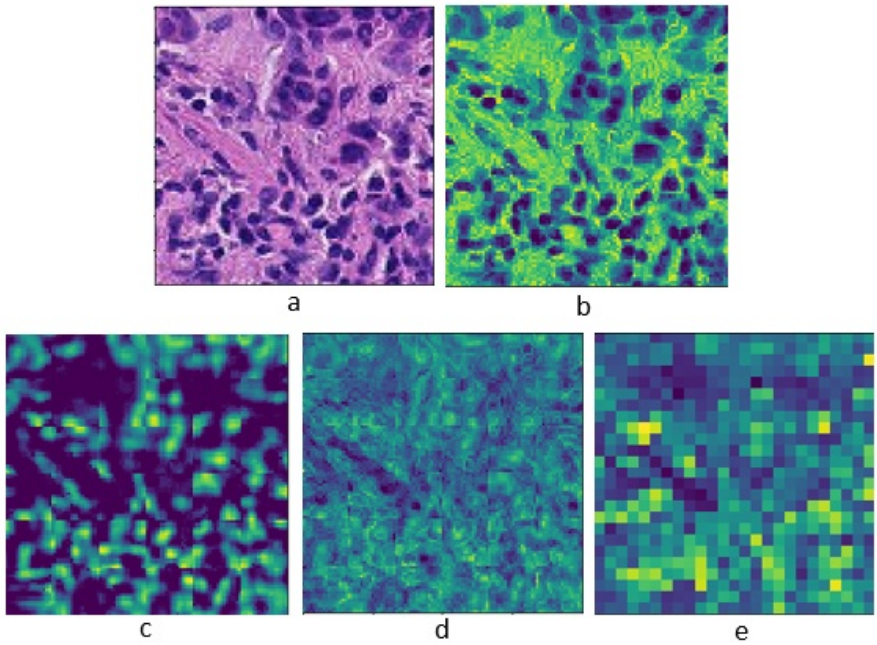


Figure 5.10: a) Input, b)-d) Extracted Features, e) Heatmap

the two models were still comparable, namely 0.91 and 0.87.

Model	TP	FN	TN	FP	Acc	Pre	Rec
VGG19	361	29	206	28	0.91	0.88	0.88
InceptionV3	345	45	179	55	0.84	0.80	0.76
MobileNet	370	20	195	39	0.91	0.90	0.95
Majority Voting	370	20	198	36	0.91	0.91	0.85
Interconnected	383	7	177	57	0.90	0.87	0.98

Table 5.2: Confusion matrix and classification report of chest X-ray dataset.

The model was also tested on a malaria test set that had never been used before in the training process. Table 5.3 shows slightly

different results where two of the single models achieved good results while one model had lower performance than the previous two models. Models VGG19 and InceptionV3 achieved an accuracy value of 0.87, while MobileNet achieved an accuracy value of 0.72. MobileNet's ability was not optimum due to the significant prediction errors when predicting images containing malaria. Thus these results brought out two dominant models in the voting process. So that the accuracy value of Majority Voting can still be maintained at a reasonably good value, namely 0.86. On the other hand, the Interconnected model worked adequately well by assigning precise weights to every single network, raising the accuracy value to 0.88. This can be seen in detail in Table 5.3 where the Interconnected Model reached a recall value of up to 0.82 compared to the Majority Voting, which only reaches 0.74. These values were obtained from the number of images that the two models can predict accurately. The Interconnected Model accurately predicted 823 malaria images with 177 errors compared to the Majority Voting, which was only able to accurately predict as many as 737 malaria images with 263 prediction errors. Even so, the Precision value of all models was above 0.93.

The final evaluation process involved the PatchCam test set, which has never been used before by all models in the training process. From Table 5.4, it can be seen that this experiment only shows

Model	TP	FN	TN	FP	Acc	Pre	Rec
VGG19	798	202	938	62	0.87	0.93	0.80
InceptionV3	776	224	963	37	0.87	0.95	0.78
MobileNet	443	557	997	3	0.72	0.99	0.44
Majority Voting	737	263	988	12	0.86	0.98	0.74
Interconnected	823	177	942	58	0.88	0.93	0.82

Table 5.3: Confusion matrix and classification report of Malaria dataset.

one dominant model, namely the VGG19 model with an accuracy value of 0.87. Meanwhile, the other two models achieved accuracy values of 0.73 for InceptionV3 and 0.66 for MobileNet. This result influenced the voting process, which was only able to reach a score of 0.83. Concurrently, the Interconnected Model achieved an accuracy value of 0.87, proving that the Interconnected Model can find the correct weight for every single network in the training process. It was also verified by the Precision and Recall value of the Interconnected Model, which was better than the Majority Voting value, namely, 0.89 compared to 0.88 for the Precision value and 0.77 compared to 0.66 the recall value.

5.7. Discussion

After going through several experiments involving all three datasets, we can conclude that the Interconnected Model was a re-

Model	TP	FN	TN	FP	Acc	Pre	Rec
VGG19	6692	2199	12400	709	0.87	0.89	0.75
InceptionV3	6638	2253	8838	4271	0.70	0.61	0.75
MobileNet	2696	6195	11882	1227	0.66	0.69	0.30
Majority Voting	5902	2989	12342	767	0.83	0.88	0.66
Interconnected	6721	2170	12316	793	0.87	0.89	0.77

Table 5.4: Confusion matrix and classification report of PatchCam dataset.

liable tool when the accuracy of majority voting was low, proven by looking at the results of experiments involving the Malaria and PatchCam dataset. In these two experiments, the interconnected model's performance was slightly better than Majority Voting due to the presence of at least one model that was not well performed. Meanwhile, in the first experiment using the chest X-ray dataset, the entire single network reached the optimum value, increasing the accuracy value of the majority voting. The Interconnected Model also made better predictions than majority voting in positive classes or images containing the disease. The way the interconnected model works, which gives weight appropriately to each subnetwork, was proven to work more efficiently than initially training the three models to determine the ranking and then assign the appropriate weight according to the achievement of each sub-network.

This work focused on investigating the work efficiency of the Interconnected Model and not on optimizing the accuracy value of

each network. In other words, the use of transfer learning in this work did not involve the fined-tunning process, which can be seen from the achievement of the accuracy value of the proposed model with several similar works using the same dataset [29, 33, 42].

6. Conclusion

In this dissertation, we have shown several methods that can be used in classifying medical images. The first method was utilizing the CNN single network to classify medical images, which was proven that a CNN network could be designed by adopting the pre-trained VGG19 model. The resulting model has proven reliable for detecting cancer cells in human histopathology images with an accuracy score of 0.92.

Although it achieved quite satisfying results in detecting cancer cells on the histopathology image, the value of a single network's accuracy was still low compared to the accuracy value of the ensemble network on several related works. Therefore, on our second method, we proposed an ensemble network built from scratch, consisting of three single networks. The voting method was chosen to decide the ensemble network's final results. This method was proven to increase the accuracy value up to 0.96 when predicting cancer cells on human histopathology images.

In the decision-making process, it was found that there was a possibility that the majority voting method may lead to a false decision. If two models agreed on the false decision, it can be sure that the final decision would be false, which will reduce the accuracy value of the ensemble network. Hence, the interconnected model was proposed, which was architecturally similar to ensemble

networks, although it did not use the voting method in determining the final result. The interconnected network that trained all three single networks simultaneously was proven to assign proper weights to all three single networks for its final decision. Thus, an interconnected network can be used as an option when the accuracy value of the ensemble network based on the majority voting score is low.

Summary

This dissertation was motivated by several things. The first one was the development of technology that allows digitizing histopathological images leading to a massive explosion of data which allows the reuse processing. In line with this, the increasing ability of machines to perform computational processes is an opportunity to process large amounts of data. Based on this background, this dissertation was focused on predicting medical images using Convolutional Neural Network (CNN).

There were three architectural approaches proposed to build the model; the first architecture was using a single CNN network. This VGG modules relied on a stack of convolutional blocks. In detail, this model can be illustrated by stacking eight convolution layers, each with two convolution layers with 64 channels, two 128 channels convolution layers, two 256 channels convolution layers, and two 512 channels convolution layers. Next, a Max-Pooling layer was inserted in each of the two convolution layers. Thus, the architecture will form four convolutional blocks, with each block having two convolutional layers and a pooling layer. Architecture with a Multilayer Perceptron (MLP) filled with a Flatten layer which functions to change the dimensions of the feature. Next, two Fully Connected layers were installed, each of which has 64 neurons, and a Fully Connected layer with the Sigmoid activation

function was installed as the last layer. This experiment succeeded in achieving a value of 0.92 and Area Under Curve 0.98 in the Receiving Operator Characteristics Curve.

The second approach was to build a series of models by combining three well-known modules: VGG, Inception, and ResNet. The final result was then determined by conducting a voting process. The first voting process was majority voting, where the three sub-networks have equal weight to make decisions. By having the same weight, if two models agree on one decision and one model with different decisions, the two models' decisions will be taken. The second voting process was to give the appropriate weight to the sub-networks abilities when trained separately. With this voting method, the sub-networks that achieve better accuracy will have a greater weight than the sub-networks that have less accuracy score. The results achieved from the voting process using both the majority vote method and the weighted majority vote reached 0.96.

The final approach was based on the idea of putting weighting tasks as part of the training process. As a result, neither at the start nor at the end of the training period, the user calculated the weighting. Technically, it can be described that the architecture was built in an interconnected model. The interconnected model consisted of three sub-models that trained together to predict med-

ical images. By training together, the training process itself determines which model will affect the overall model. Training the three models together impacts the number of training parameters where all the model parameters were trained together. For this reason, considering machine capability and computational costs, the pre-trained model and transfer learning method were used. Technically, the three models were three well-known trained models, namely VGG19, InceptionV3, and MobileNet. In this experiment, three different data sets were used to maintain the objectivity of the research results, namely: the X-ray dataset, the Malaria dataset, and the PatchCam dataset, which were also used in the two previous studies. Then as a final step, the accuracy value of the interconnected model was compared with the ensemble majority voting.

From the experimental results using the three datasets, different results were found. In the X-ray dataset, the three sub-models have equal results. These results improved the majority vote's accuracy score, which reached 0.91, while the interconnected model score was 0.90. On the other hand, the experiment using the Malaria dataset showed that the three sub-models did not show same level of accuracy, with VGG19 and InceptionV3 better then MobileNet. Thus, the two sub-models influenced the voting process more than the other sub-models, increasing the accuracy value of the model to 0.86. Likewise, the interconnected model can main-

tain the consistency of the accuracy value up to 0.88. The third experiment used the PatchCam dataset; this experiment gave three different results for the sub-model 0.87, 0.70, 0.66 respectively for VGG19, InceptionV3, and MobileNet. This result affected the accuracy value of the majority voting, which was 0.83 compared to the accuracy value of the interconnected models, which remained consistent at 0.87. Another result obtained was that although the accuracy value of the interconnected model was slightly different from the ensemble majority voting model, the interconnected model was constantly better than the ensemble majority model in predicting data in positive classes, which in the context of these three datasets were images that were indicated to have disease.

In the end, this dissertation concluded that the CNN ensemble network performed better than the single CNN network in the case of predicting medical images. When the accuracy of the ensemble majority voting model was low, the interconnected model can be considered as a solution.

Publications

Journal publications related to this dissertation:

1. Oktavian Lantang, Gyorgy Terdik, Andras Hajdu, and Attila Tiba. "Comparison of single and ensemble-based convolutional neural network for cancerous image classification". In: *Annales Mathematicae Et Informaticae* 54 (2021), pp. 45-56.

2. Oktavian Lantang, Gyorgy Terdik, Andras Hajdu, and Attila Tiba. "Investigation of the efficiency of an interconnected convolutional neural network by classifying medical images". In: *Annales Mathematicae et Informaticae* 53 (2021), pp. 219-234.

Conference and proceedings related to this dissertation:

1. Oktavian Lantang, Attila Tiba, Andras Hajdu, and Gyorgy Terdik. "Convolutional Neural Network for Predicting The Spread of Cancer". In: *Proceedings of the 2019 10th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*. Naples, Italy: IEEE, 2019, pp. 175-180.

References

[1] Syed Muhammad Anwar, Muhammad Majid, Adnan Qayyum, Muhammad Awais, Majdi Alnowami, and Muhammad Khurram Khan. "Medical image analysis using convolutional neural networks: a review". In: *Journal of Medical Systems* 42.11 (2018), pp. 1–13. DOI: https://doi.org/10.1007/s10916-018-1088-1.

[2] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes Van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen AWM Van Der Laak, and CAMELYON16 Consortium. "Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer". In: *JAMA* 318.22 (2017), pp. 2199–2210. DOI: http://dx.doi.org/10.1001/jama.2017.14585.

[3] Léon Bottou. "Large-scale machine learning with stochastic gradient descent". In: *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186. DOI: http://dx.doi.org/10.1007/978-3-7908-2604-3_16.

[4] Léon Bottou. "Stochastic gradient descent tricks". In: *Neural Networks: Tricks of The Trade*. Springer, 2012, pp. 421–436. DOI: http://dx.doi.org/10.1007/978-3-642-35289-8_25.

[5] Freddi Bray, Jacques Ferlay, Isabelle Soerjomataram, Rebecca Siegel, Lindsey Torre, and Ahmedin Jemal. "Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries". In: *CA: A Cancer Journal for Clinicians* 68.6 (2018), pp. 394–424. DOI: https://doi.org/10.3322/caac.21492.

[6] Peter Burai, Andras Hajdu, Felipe-Riverón Edgardo Manuel, and Balazs Harangi. "Segmentation of the uterine wall by an ensemble of fully convolutional neural networks". In: *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2018, pp. 49–52. DOI: http://dx.doi.org/10.1109/EMBC.2018.8512245.

[7] Francois Chollet. In: *Deep learning with pyhton*. Manning Publication Co, 2018, p. 160.

[8] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. "Multi-column deep neural networks for image classification". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pp. 3642–3649. DOI: http://dx.doi.org/10.1109/CVPR.2012.6248110.

[9] George E Dahl, Tara N Sainath, and Geoffrey E Hinton. "Improving deep neural networks for LVCSR using rectified lin-

ear units and dropout". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2013, pp. 8609–8613. DOI: http://dx.doi.org/10.1109/ICASSP.2013.6639346.

[10] Bob D De Vos, Jelmer M Wolterink, Pim A De Jong, Max A Viergever, and Ivana Išgum. "2D image classification for 3D anatomy localization: employing deep convolutional neural networks". In: *Medical Imaging 2016: Image Processing*. Vol. 9784. International Society for Optics and Photonics. 2016, 97841Y. DOI: http://dx.doi.org/10.1117/12.2216971.

[11] Michal Drozdzal, Eugene Vorontsov, Gabriel Chartrand, Samuel Kadoury, and Chris Pal. "The importance of skip connections in biomedical image segmentation". In: *Deep Learning and Data Labeling for Medical Applications*. Springer, 2016, pp. 179–187. DOI: http://dx.doi.org/10.1007/978-3-319-46976-8_19.

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. "Deep feedforward networks". In: *Deep learning*. USA: MIT press, 2016, p. 181. URL: https://www.deeplearningbook.org/.

[13] Jane Grossman, Michael Grossman, and Robert Katz. In: *The first system of weighted differential and integral calculus*. Non-Newtonian Calculus, 2006.

[14] Balasz Harangi, Agnes Baran, and Andras Hajdu. "Classification of skin lesions using an ensemble of deep neural networks". In: *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society(EMBC)*. Honolulu, HI, USA: IEEE, 2018, pp. 2575–2578. DOI: https://doi.org/10.1109/EMBC.2018.8512800.

[15] Balazs Harangi. "Skin lesion classification with ensembles of deep convolutional neural networks". In: *Journal of biomedical informatics* 86 (2018), pp. 25–32. DOI: http://dx.doi.org/10.1016/j.jbi.2018.08.006.

[16] Balazs Harangi, Janos Toth, and Andras Hajdu. "Fusion of deep convolutional neural networks for microaneurysm detection in color fundus images". In: *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2018, pp. 3705–3708. DOI: http://dx.doi.org/10.1109/EMBC.2018.8513035.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the 2016 IEEE Conference on Computer Vision and*

Pattern Recognition. Las Vegas, NV, USA: IEEE, 2016, pp. 770–778. DOI: https://doi.org/10.1109/CVPR.2016.90.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Delving deep into rectifiers: surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE International Conference on Computer Vision*. Santiago,Chile: IEEE, 2015, pp. 1026–1034. DOI: https://doi.org/10.1109/ICCV.2015.123.

[19] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. "Bag of tricks for image classification with convolutional neural networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 558–567. DOI: http://dx.doi.org/10.1109/CVPR.2019.00065.

[20] Mohamed Hosni, Ibtissam Abnane, Ali Idri, Juan M. Carillo de Gea, and Jose Luis Fernandez Aleman. "Reviewing ensemble classification methods in breast cancer". In: *Computer Methods and Programs in Biomedicine* 177 (2019), pp. 89–112. DOI: https://doi.org/10.1016/j.cmpb.2019.05.019.

[21] Wei Hu, Yangyu Huang, Li Wei, Fan Zhang, and Hengchao Li. "Deep convolutional neural networks for hyperspectral image

classification". In: *Journal of Sensors* 2 (2015), pp. 1–12. DOI: http://dx.doi.org/10.1155/2015/258619.

[22] Sara Hosseinzadeh Kassani, Peyman Hosseinzadeh Kassani, Michael J. Wesolowski, Kevin A. Schneider, and Ralph Deters. "Classification of hispatology biopsy images using ensemble of deep learning networks". In: *arXiv preprint arXiv:1909.11870* (2019).

[23] Brady Kieffer, Morteza Babaie, Shivam Kalra, and Hamid R Tizhoosh. "Convolutional neural networks for histopathology image classification: Training vs. using pre-trained networks". In: *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE, pp. 1–6. DOI: https://doi.org/10.1109/IPTA.2017.8310149.

[24] Diederik P. Kingma and Jimmy Lei Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. "ImageNet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (2017), pp. 1079–1105. DOI: https://doi.org/10.1145/3065386.

[26] Ashnil Kumar, Jinman Kim, David Lyndon, Michael Fulham, and Dagan Feng. "An ensemble of fine-tuned convolutional

neural networks for medical image classification". In: *IEEE Journal of Biomedical and Health Informatics* 21.1 (2016), pp. 31–40. DOI: http://dx.doi.org/10.1109/JBHI.2016.2635663.

[27] Oktavian Lantang, Gyorgy Terdik, Andras Hajdu, and Attila Tiba. "Comparison of single and ensemble-based convolutional neural networks for cancerous image classification". In: *Annales Mathematicae et Informaticae* 54 (2021), pp. 45–56. DOI: https://doi.org/10.33039/ami.2021.03.013.

[28] Oktavian Lantang, Gyorgy Terdik, Andras Hajdu, and Attila Tiba. "Investigation of the efficiency of an interconnected convolutional neural network by classifying medical images". In: *Annales Mathematicae et Informaticae* 53 (2021), pp. 219–234. DOI: https://doi.org/10.33039/ami.2021.04.001.

[29] Oktavian Lantang, Attila Tiba, Andras Hajdu, and Gyorgy Terdik. "Convolutional neural network for predicting the spread of cancer". In: *Proceedings of the 2019 10th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*. Naples, Italy: IEEE, 2019, pp. 175–180. DOI: https://doi.org/10.1109/CogInfoCom47531.2019.9089939.

[30] Yann LeCun and Yoshua Bengio. “Convolutional networks for images, speech, and time series”. In: *The Handbook of Brain Theory and Neural Networks* 3361.10 (1995).

[31] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *Nature* 521 (2015), pp. 436–444. DOI: https://doi.org/10.1038/nature14539.

[32] Qing Li, Weidong Cai, Xiaogang Wang, Yun Zhou, David Dagan Feng, and Mei Chen. “Medical image classification with convolutional neural network”. In: *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*. IEEE. 2014, pp. 844–848. DOI: https://doi.org/10.1109/ICARCV.2014.7064414.

[33] Zhaohui Liang, Andrew Powell, Ilker Ersoy, Mahdieh Poostchi, Kamolrat Silamut, Kannappan Palaniappan, Peng Guo, Md Amir Hossain, Antani Sammer, Richard James Maude, Jimmy Xiangji Huang, Stefan Jaeger, and George Thoma. “CNN-based image analysis for malaria diagnosis”. In: *2016 IEEE Conference on Bioinformatics and Biomedicine (BBIM)*. Shenzhen, China: IEEE, 2016, pp. 493–496. DOI: https://doi.org/10.1109/BIBM.2016.7822567.

[34] Le Lu, Yefeng Zheng, Gustavo Carneiro, and Lin Yang. “Deep learning and convolutional neural networks for medical image

computing". In: *Advances in Computer Vision and Pattern Recognition*. Springer, 2017.

[35] Anant Madabhushi. "Digital pathology image analysis: opportunities and challenges". In: *Imaging in Medicine* 1.1 (2009), pp. 7–10. DOI: https://doi.org/10.7282/T38C9TMS.

[36] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. "Convolutional neural networks for large-scale remote-sensing image classification". In: *IEEE Transactions on Geoscience and Remote Sensing* 55.2 (2016), pp. 645–657. DOI: http://dx.doi.org/10.1109/TGRS.2016.2612821.

[37] Radko Mesiar and Jana Spirkova. "Weighted means and weighting functions". In: *Kybernetika* 42.2 (2006), pp. 151–160. URL: https://dml.cz/handle/10338.dmlcz/135706.

[38] Agnieszka Mikolajczyk and Michal Grochowski. "Data augmentation for improving deep learning in image classification problem". In: *2018 International Interdisciplinary PhD Workshop(IIPhDW)*. Swinoujście, Poland: IEEE, 2018. DOI: https://doi.org/10.1109/IIPHDW.2018.8388338.

[39] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. "V-net: Fully convolutional neural networks for volumetric medical image segmentation". In: *2016 Fourth International*

Conference on 3D Vision (3DV). IEEE. 2016, pp. 565–571. DOI: https://doi.org/10.1109/3DV.2016.79.

[40] Chigozie Enyinna Nwankpa, Wnifred Ijomah, Anthony Gachagan, and Stephan Marshall. "Activation functions: Comparison of trend in practice and research for deep learning". In: *arXiv preprint arXiv:1811.03378* (2018).

[41] Kuntal Kumar Pal and KS Sudeep. "Preprocessing for image classification by convolutional neural networks". In: *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. IEEE. 2016, pp. 1778–1781. DOI: http://dx.doi.org/10.1109/RTEICT.2016.7808140.

[42] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Robyn L. Ball, Curtis Langlotz, Katie Shpanskaya, Matthew P. Lungren, and Andrew Y. Ng. "CheXNet: Radiologist-level pneumonia detection on chest x-rays with deep learning". In: *arXiv preprint arXiv:1711.05225* (2017).

[43] Prajit Ramachandran, Barret Zoph, and Quoc V Le. "Searching for activation functions". In: *arXiv preprint arXiv:1710.05941* (2017).

[44] Waseem Rawat and Zenghui Wang. "Deep convolutional neural networks for image classification: A comprehensive review". In: *Neural Computation* 29.9 (2017), pp. 2352–2449. DOI: https://doi.org/10.1162/neco_a_00990.

[45] Peter Sadowski. *Notes on backpropagation*. 2016. URL: https://www.ics.uci.edu/~pjsadows/notes.pdf.

[46] Benedetta Savelli, Alessandro Bria, Mario Molinara, Claudio Marrocco, and Francesco Tortorella. "A multi-context CNN ensemble for small lesion detection". In: *Artificial Intelligence in Medicine* 103 (2020), pp. 1–13. DOI: https://doi.org/10.1016/j.artmed.2019.101749.

[47] Neha Sharma, Vibhor Jain, and Anju Mishra. "An analysis of convolutional neural networks for image classification". In: *Procedia Computer Science* 132 (2018), pp. 377–384. DOI: https://doi.org/10.1016/j.procs.2018.05.198.

[48] Hoo-Chang Shin, Holger Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald Summers. "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning". In: *IEEE Transactions on Medical Imaging* 35.5 (2016), pp. 1285–1298. DOI: https://doi.org/10.1109/tmi.2016.2528162.

[49] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* ().

[50] Viktor Slavkovikj, Steven Verstockt, Wesley De Neve, Sofie Van Hoecke, and Rik Van de Walle. “Hyperspectral image classification with convolutional neural networks”. In: *Proceedings of the 23rd ACM International Conference on Multimedia*. 2015, pp. 1159–1162. DOI: http://dx.doi.org/10.1145/2733373.2806306.

[51] Yanan Sun, Bing Xue, Mengjie Zhang, and Gary G Yen. “Evolving deep convolutional neural networks for image classification”. In: *IEEE Transactions on Evolutionary Computation* 24.2 (2019), pp. 394–407. DOI: http://dx.doi.org/10.1109/TEVC.2019.2916183.

[52] Christian Szegedy, Wei Liu, Yangqing Jia, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. “Going deeper with convolutions”. In: *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*. Boston, MA, USA: IEEE, 2015, pp. 1–9. DOI: https://doi.org/10.1109/cvpr.2015.7298594.

[53] Nima Tajbakhsh, Jae Y Shin, Suryakanth R Gurudu, R Todd Hurst, Christopher B Kendall, Michael B Gotway, and Jian-

ming Liang. “Convolutional neural networks for medical image analysis: Full training or fine tuning?” In: *IEEE Transactions on Medical Imaging* 35.5 (2016), pp. 1299–1312. DOI: http://dx.doi.org/10.1109/TMI.2016.2535302.

[54] Lisa Torrey and Jude Shavlik. “Transfer learning”. In: *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*. IGI global, 2010, pp. 242–264.

[55] Bastiaan S. Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. “Rotation equivariant CNNs for digital pathology”. In: *International Conference on Medical Image Computing and computer-Assisted Intervention*. Springer, Cham, 2018, pp. 210–218. URL: https://link.springer.com/chapter/10.1007/978-3-030-00934-2_24.

[56] Robert Weinberg. “How cancer arises”. In: *Scientific American* 275.3 (1996), pp. 62–70. DOI: http://dx.doi.org/10.1038/scientificamerican0996-62.

[57] Ralph Weissleder. “Molecular imaging in cancer”. In: *Science* 312.5777 (2006), pp. 1168–1171. DOI: https://doi.org/10.1126/science.1125949.

[58] Tian Xia, Ashnil Kumar, Dagan Feng, and Jinman Kim. “Patch-level tumor classification in digital hispatology im-

ages with domain adapted deep learning". In: *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Honolulu, HI, USA: IEEE, 2018, pp. 644–647. DOI: http://dx.doi.org/10.1109/EMBC.2018.8512353.

[59] Samir S Yadav and Shivajirao M Jadhav. "Deep convolutional neural network based medical image classification for disease diagnosis". In: *Journal of Big Data* 6.1 (2019), pp. 1–18. URL: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0276-2.

[60] M.D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q.V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G.E. Hinton. "On rectified linear units for speech processing". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013, pp. 3517–3521. DOI: https://doi.org/10.1109/ICASSP.2013.6638312.